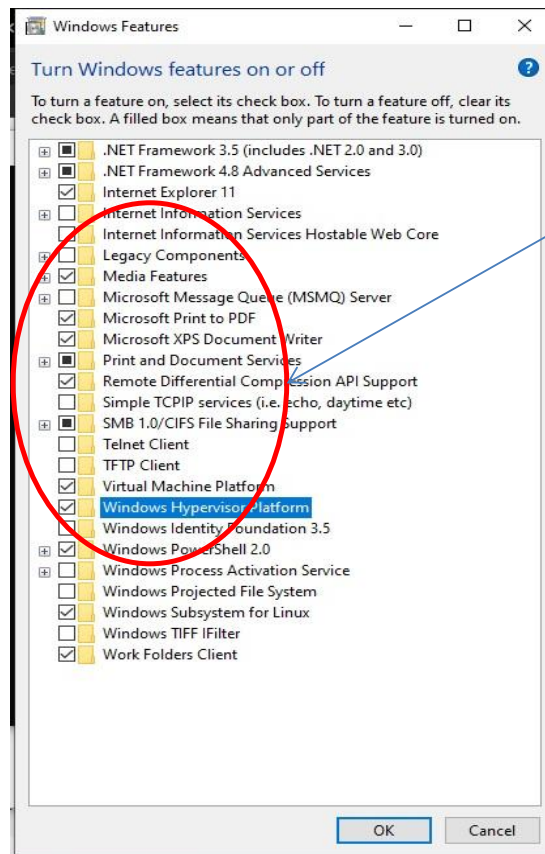


	<p>Prerequisites</p> <p>Before I go any further,K8 Deployment will strongly recommend a machine with at least 8G of RAM. Some of the following options are based on virtual machines (VM), current operating systems require at least 4GB of RAM before any issues would happen. Having 8GB of RAM allows for options like VirtualBox to get sufficient resources to be able to do more than the bare minimum.</p> <p>Then, we could also install kubectl on local machine. It’s the common interface with any Kubernetes cluster, and local instance is no different. I will say that you could call straight to the API, but I’d only advise doing this if we had a specific reason to.</p> <p>Specification:</p> <p><i>OS Name</i> : Microsoft Windows 10 Home Single Language <i>Version</i> : 10.0.18363 Build 18363 <i>System Type</i> : x64-based PC <i>Processor</i> : Intel(R) Pentium(R) Silver N5000 CPU @ 1.10GHz, 1101 Mhz, 4 Core(s), 4 Logical Processor(s) <i>BIOS Mode</i> : UEFI <i>Virtualization</i> : ENABLED</p> <p>Installed Softwares & Plugins</p> <ol style="list-style-type: none">1. Minikube (v1.18.1)2. Kubernetes CLI (v1.20.5)3. DOCKER (for Desktop v3.2.2)4. CHOCOLATEY (v 10.15)5. WSL (Windows Subsystem for Linux) x646. Virtualization Enabled (Hyper-V)
2	<p>INSTALLATION PROCEDURE</p> <p>A. CONFIGURE SYSTEM</p> <p>a.1 Turn ON Hyper-V (Virtualization) "ENABLED"</p>



**Hyper-V is not
AVAILABLE**

* Install Hyper-V on Windows 10

* Enable Hyper-V to create virtual machines on Windows 10.

Hyper-V can be enabled in many ways including using the Windows 10 control panel, PowerShell or using the Deployment Imaging Servicing and Management tool (DISM).

Note: Hyper-V is built into Windows as an optional feature -- there is no Hyper-V download.

* Enable Hyper-V using PowerShell

Open a PowerShell console as Administrator.

Run the following command:

PowerShell:

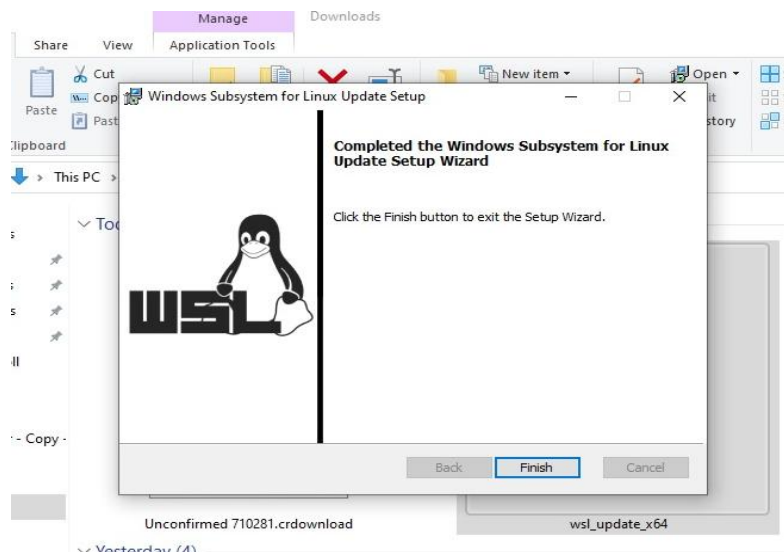
"Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All"

```
Administrator: Windows PowerShell
PS C:\> DISM /Online /Enable-Feature /All /FeatureName:Microsoft-Hyper-V
Deployment Image Servicing and Management tool
Version: 10.0.10240.16384
Image Version: 10.0.10240.16384
Enabling feature(s)
=====100.0%=====]
The operation completed successfully.
Restart Windows to complete this operation.
Do you want to restart the computer now? (Y/N) _
```

Confirmed Hyper-V Enabled

```
Administrator: Command Prompt
[08]: KB4561600
[09]: KB4565554
[10]: KB4576751
[11]: KB4577586
[12]: KB4580325
[13]: KB4586863
[14]: KB4598479
[15]: KB4601395
[16]: KB5001028
Network Card(s): 2 NIC(s) Installed.
[01]: Realtek PCIe GbE Family Controller
Connection Name: Ethernet
Status: Media disconnected
[02]: Qualcomm Atheros QCA9377 Wireless Network Adapter
Connection Name: Wi-Fi
DHCP Enabled: Yes
DHCP Server: 192.168.1.1
IP address(es)
[01]: 192.168.1.9
[02]: fe80::99f1:37f4:8373:e926
[03]: 2001:4451:8310:6b00:94bf:3f91:cad8:1bb4
[04]: 2001:4451:8310:6b00:99f1:37f4:8373:e926
Hyper-V Requirements: A hypervisor has been detected. Features required for Hyper-V will not be displayed.
C:\Windows\system32>
```

a.2 Install/Update WSL (Windows Subsystem for Linux) x64



B. SOFTWARE INSTALLATION

b1. Install "CHOCOLATEY"

Link: <https://chocolatey.org/install>

Requirements:

- Windows 7+ / Windows Server 2003+
- PowerShell v2+ (minimum is v3 for install from this website due to TLS 1.2 requirement)
- .NET Framework 4+ (the installation will attempt to install .NET 4.0 if you do not have it installed)(minimum is 4.5 for install from this website due to TLS 1.2 requirement)

```
Administrator: Command Prompt

Setting ChocolateyInstall to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'Batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at
'C:\Users\romzkie08\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
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

C:\Windows\system32>choco
```

verified CHOCOLATEY (v10.15) Installed

```
--proxy-user=VALUE
Proxy User Name - Explicit proxy user (optional). Requires explicitly
proxy ('--proxy' or config setting). Overrides the default proxy user of
''. Available for config settings in 0.9.9.9+, this CLI option available
in 0.10.4+.

--proxy-password=VALUE
Proxy Password - Explicit proxy password (optional) to be used with
username. Requires explicitly proxy ('--proxy' or config setting) and
user name. Overrides the default proxy password (encrypted in settings
if set). Available for config settings in 0.9.9.9+, this CLI option
available in 0.10.4+.

--proxy-bypass-list=VALUE
ProxyBypassList - Comma separated list of regex locations to bypass on
proxy. Requires explicitly proxy ('--proxy' or config setting). Overrides
the default proxy bypass list of ''. Available in 0.10.4+.

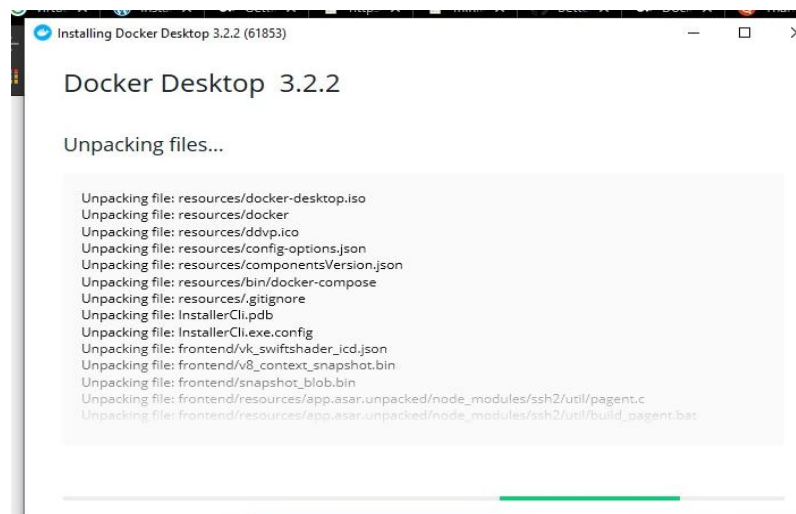
--proxy-bypass-on-local
Proxy Bypass On Local - Bypass proxy for local connections. Requires
explicitly proxy ('--proxy' or config setting). Overrides the default
proxy bypass on local setting of 'True'. Available in 0.10.4+.

--log-file=VALUE
Log File to output to in addition to regular loggers. Available in 0.1-
0.8+.

chocolatey v0.10.15

C:\Windows\system32>
```

b2. Install DOCKER (v3.2.2)



b3. Install Minikube

```
Administrator: Command Prompt

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

C:\Windows\system32>choco install minikube
Chocolatey v0.10.15
Installing the following packages:
minikube
By installing you accept licenses for the packages.
Progress: Downloading Minikube 1.18.1... 100%

Minikube v1.18.1 [Approved]
minikube package files install 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.

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Did you know the proceeds of Pro (and some proceeds from other
licensed editions) go into bettering the community infrastructure?
Your support ensures an active community, keeps Chocolatey tip top,
plus it nets you some awesome features!
https://chocolatey.org/compare

C:\Windows\system32>
```

b4. Install Kubernetes-cli

```
Administrator: Command Prompt

available in 0.10.4+.

--proxy-bypass-list=VALUE
ProxyBypassList - Comma separated list of regex locations to bypass on
proxy. Requires explicitly proxy ('--proxy' or config setting). Overrides
the default proxy bypass list of ''. Available in 0.10.4+.

--proxy-bypass-on-local
Proxy Bypass On Local - Bypass proxy for local connections. Requires
explicitly proxy ('--proxy' or config setting). Overrides the default
proxy bypass on local setting of 'True'. Available in 0.10.4+.

--log-file=VALUE
Log File to output to in addition to regular loggers. Available in 0.1-
0.8+.

Chocolatey v0.10.15

C:\Windows\system32>choco install kubernetes-cli
```

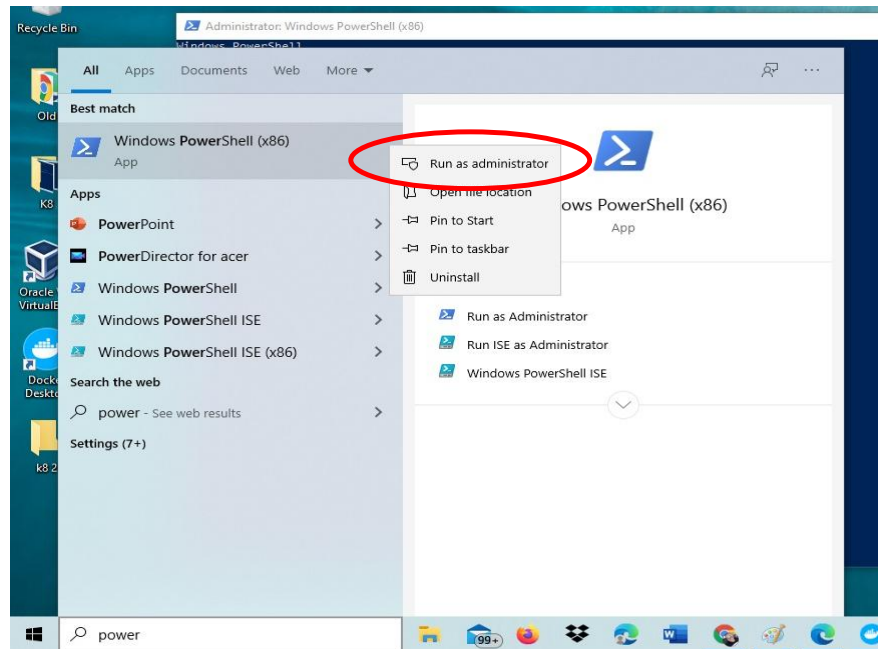
Confirm if Minikube is "working"

```
PS C:\Windows\system32> minikube start
* minikube v1.18.1 on Microsoft Windows 10 Home Single Language 10.0.18363 Build 18363
* Automatically selected the docker driver. Other choices: hyperv, virtualbox, ssh
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.20.2 on Docker 20.10.3 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32>
```

Now we can start developing the cluster using the installed programs

DEVELOPMENT PROCEDURE

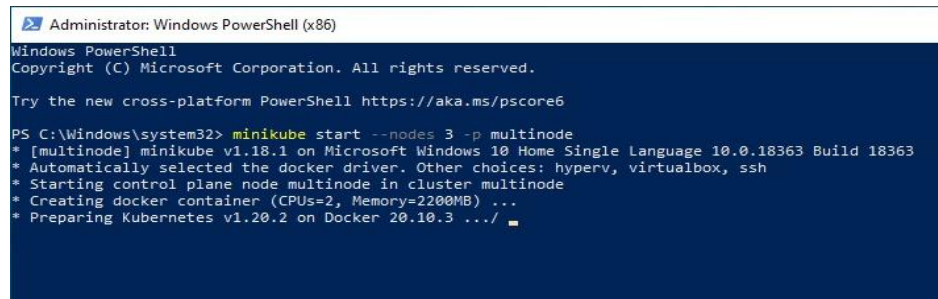
A. Use "POWERSHELL (x86)" and RUN it as "ADMINISTRATOR"



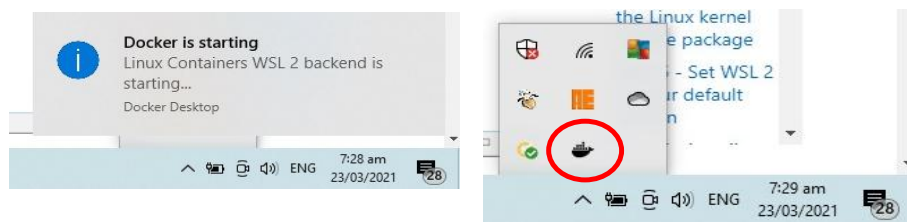
a.1 Create 3 Nodes on Minikube

CommandLine:

"Minikube start --nodes 3 -p Multinode"



a.2 Docker must "RUN" and visible on the System Tray



a.3 Minikube (MULTINODES) created 3 Nodes

- * 3 Nodes
- * 1 Master Node (Control Plane)
- * 2 Worker Node

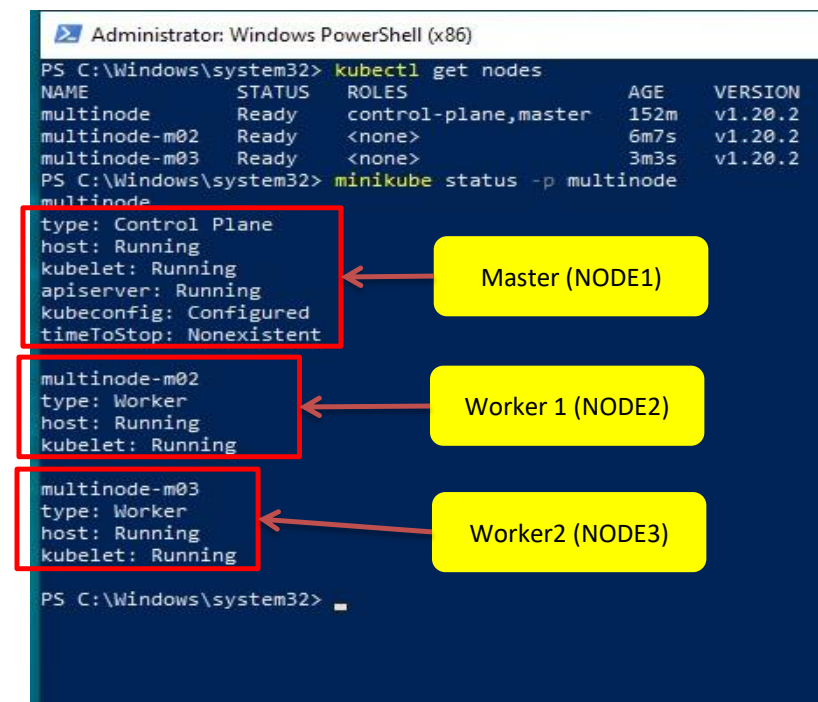
```
Administrator: Windows PowerShell (x86)
PS C:\Windows\system32> kubectl get nodes
NAME                STATUS    ROLES                  AGE      VERSION
multinode            Ready     control-plane,master   152m     v1.20.2
multinode-m02        Ready     <none>                 6m7s     v1.20.2
multinode-m03        Ready     <none>                 3m3s     v1.20.2
PS C:\Windows\system32>
```

a.3 Node Status

CommandLine:

"Minikube status -p Multinode"

```
Administrator: Windows PowerShell (x86)
PS C:\Windows\system32> kubectl get nodes
NAME                STATUS    ROLES                  AGE      VERSION
multinode            Ready     control-plane,master   152m     v1.20.2
multinode-m02        Ready     <none>                 6m7s     v1.20.2
multinode-m03        Ready     <none>                 3m3s     v1.20.2
PS C:\Windows\system32> minikube status -p multinode
multinode
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
timeToStop: Nonexistent
multinode-m02
type: Worker
host: Running
kubelet: Running
multinode-m03
type: Worker
host: Running
kubelet: Running
PS C:\Windows\system32>
```

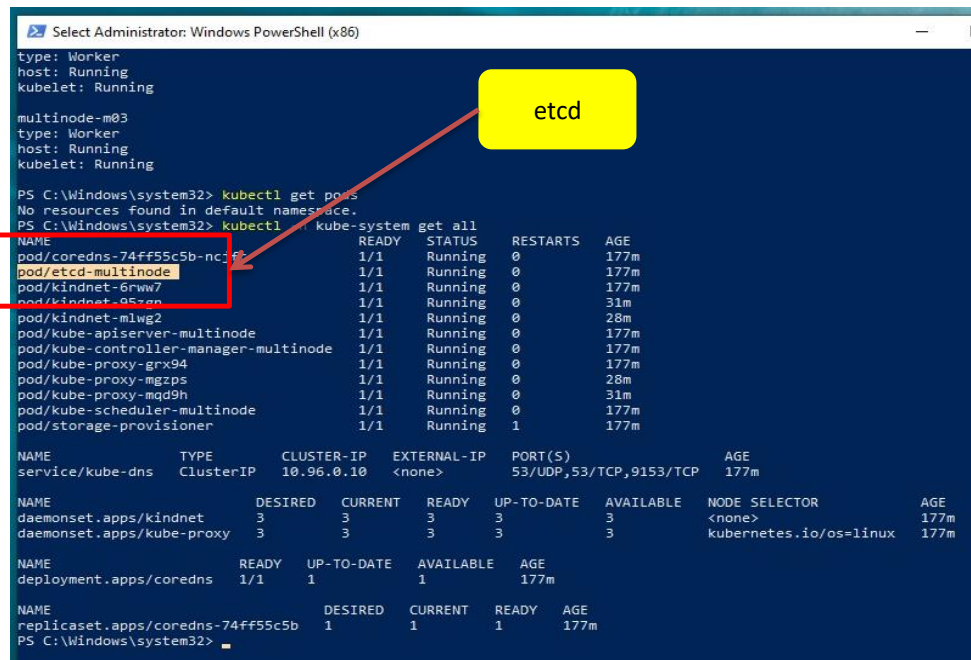


a.4 etcd created/status

CommandLine:

"*kubectl -n kube-system get all*"

<pod/etcd-multinode>



```
PS C:\Windows\system32> kubectl get pods
No resources found in default namespace.
PS C:\Windows\system32> kubectl -n kube-system get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/coredns-74ff55c5b-nc:ff	1/1	Running	0	177m
pod/etcd-multinode	1/1	Running	0	177m
pod/kindnet-6rww7	1/1	Running	0	177m
pod/kindnet-85>ao	1/1	Running	0	31m
pod/kindnet-mlwg2	1/1	Running	0	28m
pod/kube-apiserver-multinode	1/1	Running	0	177m
pod/kube-controller-manager-multinode	1/1	Running	0	177m
pod/kube-proxy-grx94	1/1	Running	0	177m
pod/kube-proxy-mgzps	1/1	Running	0	28m
pod/kube-proxy-mqd9h	1/1	Running	0	31m
pod/kube-scheduler-multinode	1/1	Running	0	177m
pod/storage-provisioner	1/1	Running	1	177m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP,53/TCP,9153/TCP	177m

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
daemonset.apps/kindnet	3	3	3	3	3	<none>	177m
daemonset.apps/kube-proxy	3	3	3	3	3	kubernetes.io/os=linux	177m

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/coredns	1/1	1	1	177m

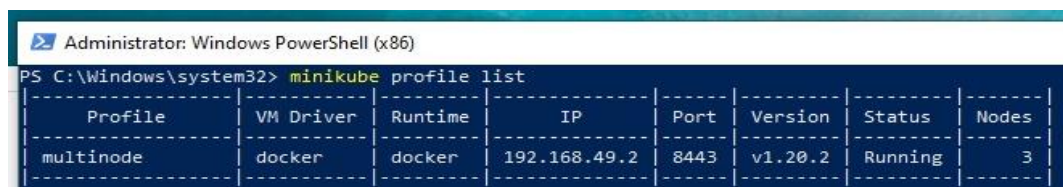
NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/coredns-74ff55c5b	1	1	1	177m

```
PS C:\Windows\system32>
```

a.5 Minikube (MULTINODE) Status

CommandLine:

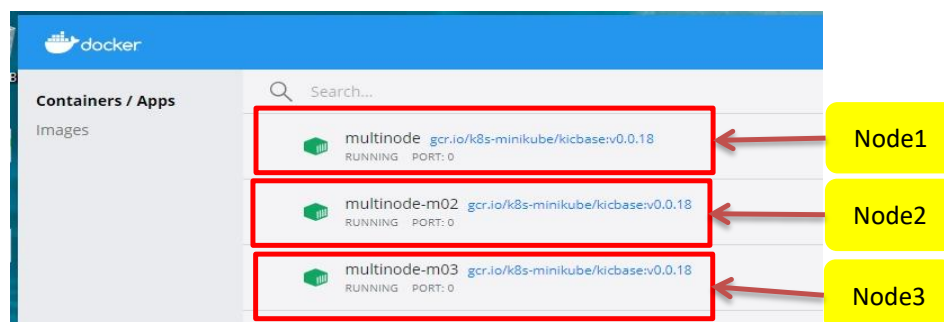
"*minikube profile list*"



```
PS C:\Windows\system32> minikube profile list
```

Profile	VM Driver	Runtime	IP	Port	Version	Status	Nodes
multinode	docker	docker	192.168.49.2	8443	v1.20.2	Running	3

a.6 Check DOCKER runs 3 Nodes



a.7 Enable Dashboard

The screenshot shows the Kubernetes Dashboard interface. The 'Nodes' tab is selected, displaying a table of cluster nodes. Three nodes are visible, each highlighted with a red box. Yellow callouts labeled 'Node1', 'Node2', and 'Node3' point to the first, second, and third nodes respectively.

Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	Memory requests (bytes)	Memory limits (bytes)	Created
multinode-m03	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	100.00m (2.50%)	100.00m (2.50%)	50.00Mi (1.64%)	50.00Mi (1.64%)	an hour ago
multinode-m02	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	100.00m (2.50%)	100.00m (2.50%)	50.00Mi (1.64%)	50.00Mi (1.64%)	an hour ago
multinode	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64	True	850.00m (21.25%)	100.00m (2.50%)	220.00Mi (7.21%)	220.00Mi (7.21%)	4 hours ago

B. CLUSTER REMOVAL AND SOFTWARE EXIT

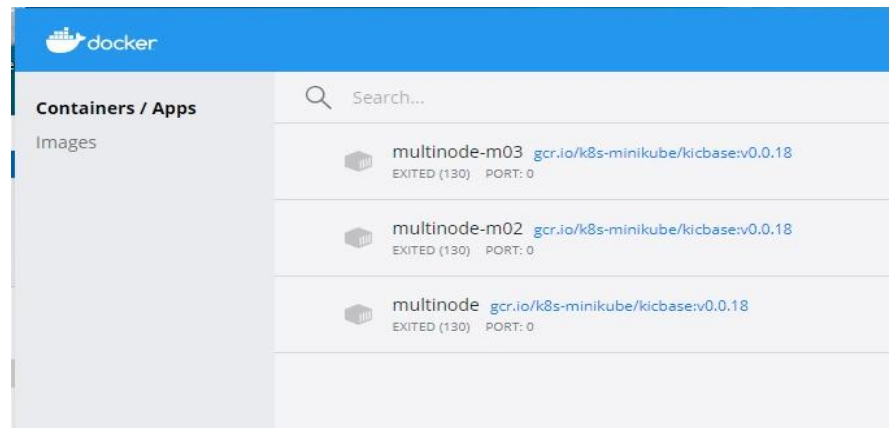
b1. Deleting CLUSTER and Container CommandLine:

"minikube delete -p multinode"

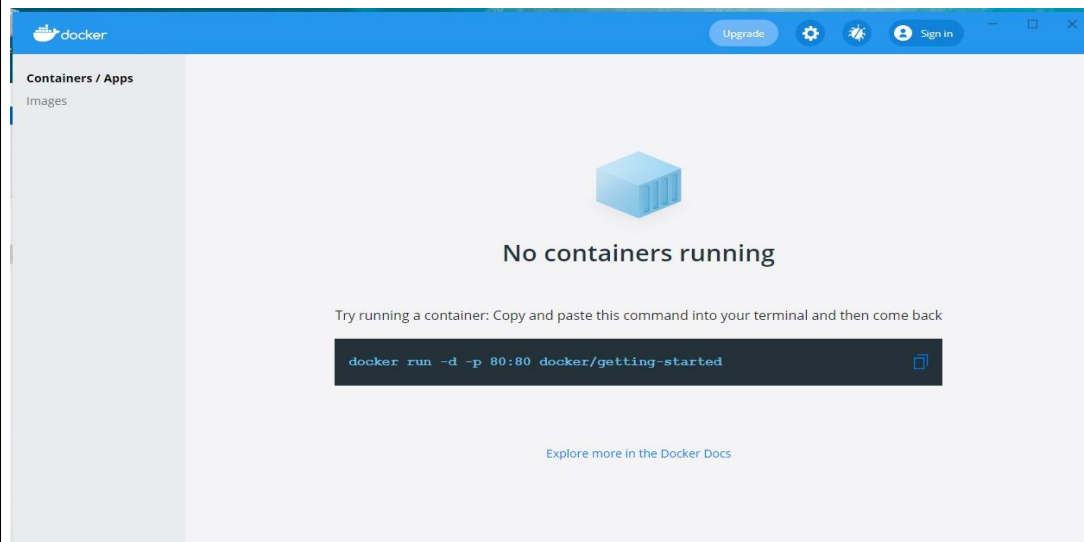
```
Administrator: Windows PowerShell (x86)

PS C:\Windows\system32> minikube delete -p multinode
* Deleting "multinode" in docker ...
* Deleting container "multinode" ...
* Deleting container "multinode-m02" ...
* Deleting container "multinode-m03" ...
* Removing C:\Users\romzkie08\.minikube\machines\multinode ...
* Removing C:\Users\romzkie08\.minikube\machines\multinode-m02 ...
* Removing C:\Users\romzkie08\.minikube\machines\multinode-m03 ...
* Removed all traces of the "multinode" cluster.
PS C:\Windows\system32> exit
```

b2. Docker > Node Exit and disabled



b3. Docker > NODE DELETED



OUTPUT

CREATION of TASK are Completed

- * 3 Nodes **DONE**
- * 1 Master **DONE**
- * 1 etcd **DONE**

all available codess and packages are availble to view at this link:

<https://github.com/rolandovillaflores/k8-demo-3nodes>