TITLE	**K8 CLUSTER DEPLOYMENT TEST**
OBJECTIVE	
1	To create and deploy K8 CLUSTER (Single Cluster)
2	Deploy ONE (1) Master, ONE (1) etcd & THREE (3) Nodes
3	Can able to get NODE info using Kubectl
NOTE:	(a) Complete within a week (b) Provide a report on how to accomplished the project
RECIPIENT	*> Reindale Ibarra <reindale.i@elabram.com>* *> fitc.hrdcha20@gmail.com*
SUMMARY	
1	System Setup & Machine Requirements
2	Installation Procedure
3	Development Procedure
4	Output
PROCESS	
1	**SYSTEM SETUP AND MACHINE REQUIREMENTS** This report describes my method to deploy Kubernetes locally to emulate what we could do in a real cloud on local machine. Running kubernetes locally can be helpful to save costs, to use portions for CI/CD on cloud-native applications, and to deal with situations when we don't have reliable internet access and want to do "real" work.

Prerequisites

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.

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.

Specification:

OS Name : Microsoft Windows 10 Home Single Language
Version : 10.0.18363 Build 18363
System Type : x64-based PC
Processor : Intel(R) Pentium(R) Silver N5000 CPU @ 1.10GHz, 1101 Mhz, 4 Core(s), 4 Logical Processor(s)
BIOS Mode : UEFI
Virtualization : ENABLED

Installed Softwares & Plugins

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) x64

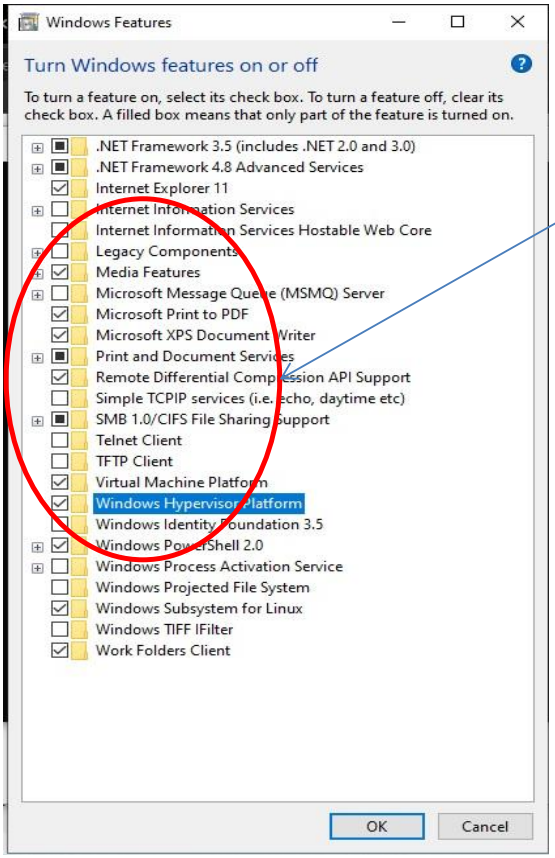
6. Virtualization Enabled (Hyper-V)

2

INSTALLATION PROCEDURE

A. CONFIGURE SYSTEM

a.1 Turn ON Hyper-V (Virtualization) "ENABLED"



Hyper-V is not AVAILABLE

* Install Hyper-V on Windows 10

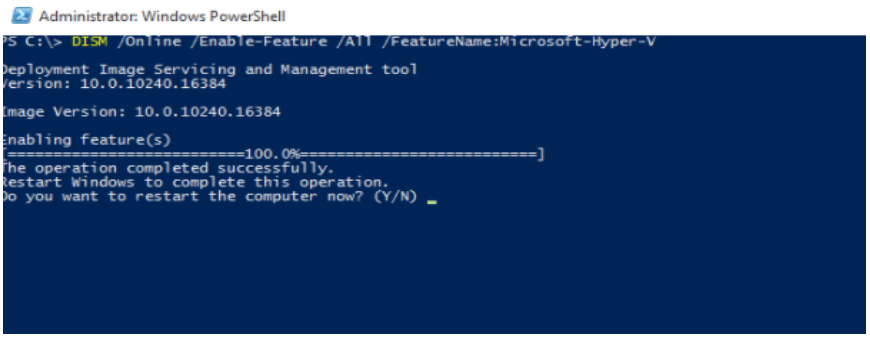
* Enable Hyper-V to create virtual machines on Windows 10.
Hyper-V can be enabled in many ways including using the Windows 10 control panel, PowerShell or using the Deployment Imaging Servicing and Management tool (DISM).

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

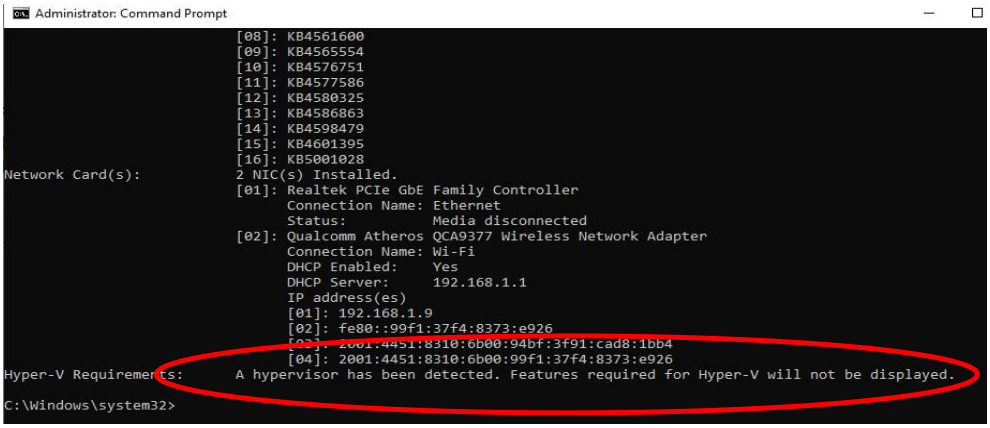
* Enable Hyper-V using PowerShell
Open a PowerShell console as Administrator.

Run the following command:

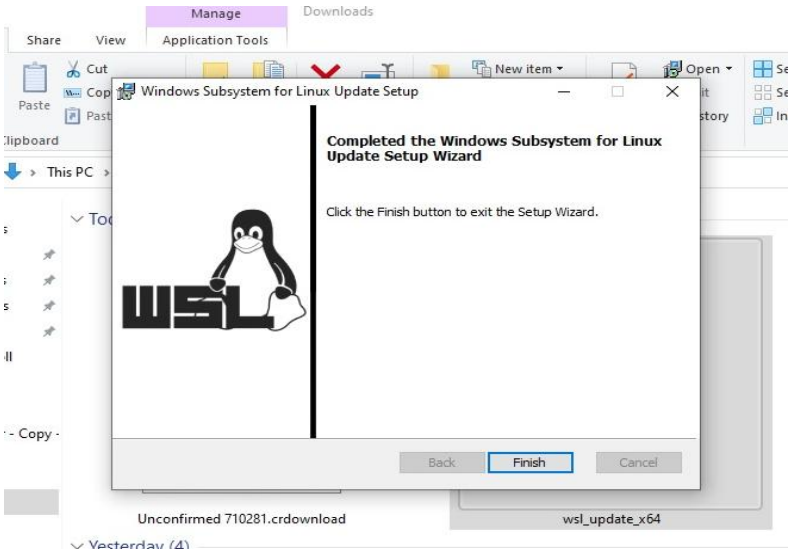
PowerShell:
"Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V -All"



Confirmed Hyper-V Enabled



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

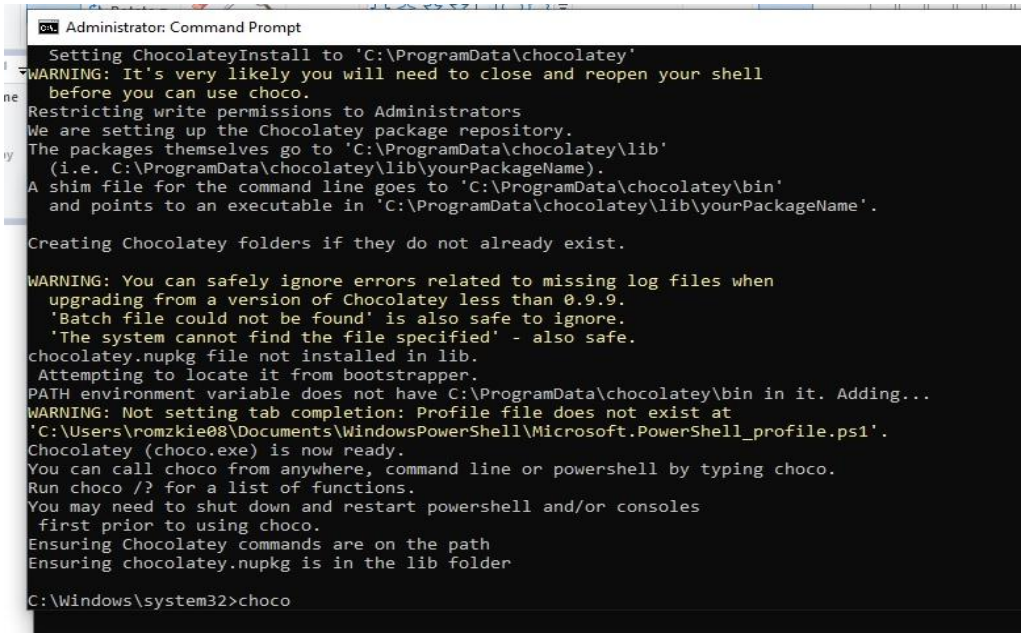


B. SOFTWARE INSTALLATION

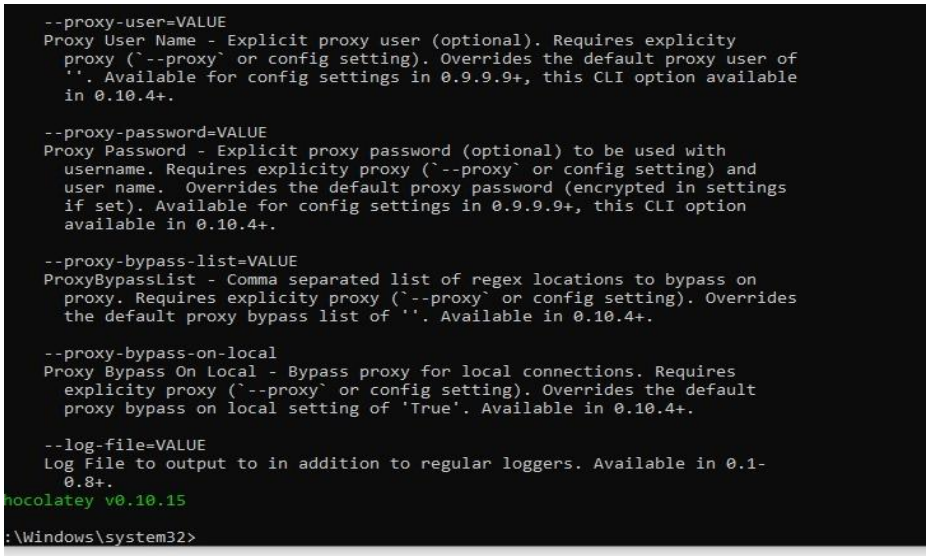
b1. Install "CHOCOLATEY"

Link: https://chocolatey.org/install

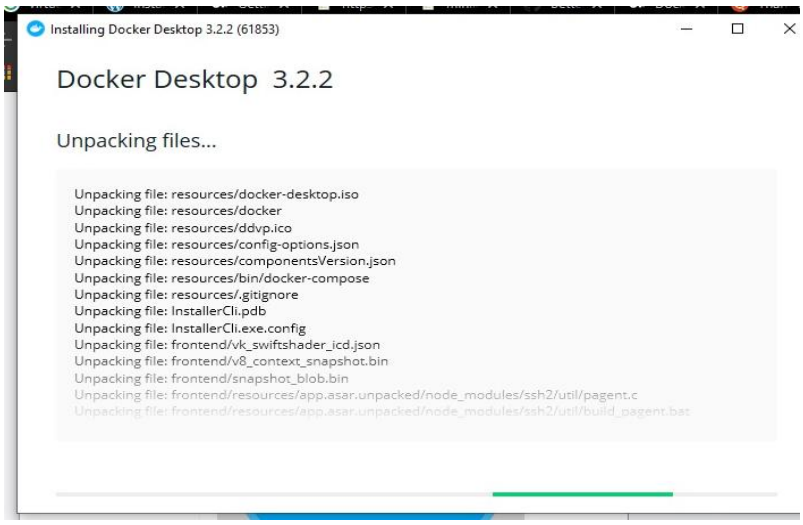
Requirements:
- Windows 7+ / Windows Server 2003+
- PowerShell v2+ (minimum is v3 for install from this website due to TLS 1.2 requirement)
- .NET Framework 4+ (the installation will attempt to install .NET 4.0 if you do not have it installed)(minimum is 4.5 for install from this website due to TLS 1.2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



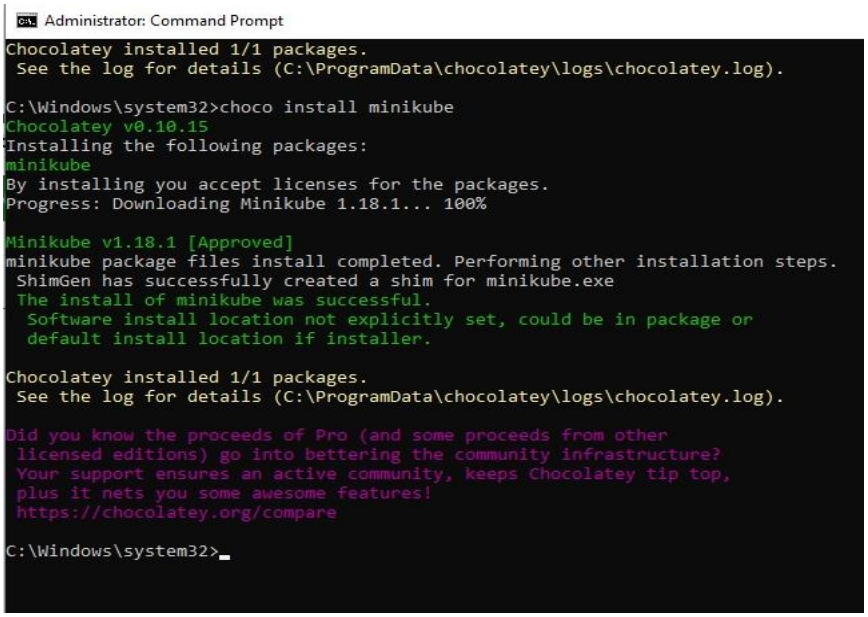
verified CHOCOLATEY (v10.15) Installed



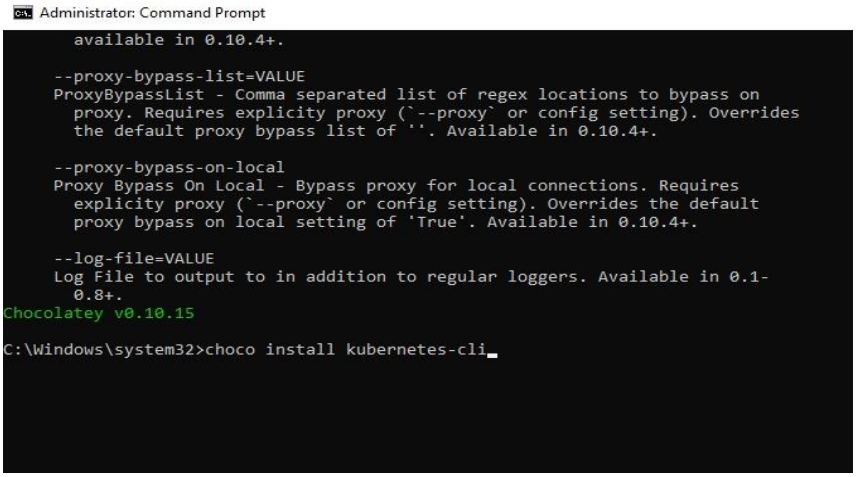
b2. Install DOCKER (v3.2.2)



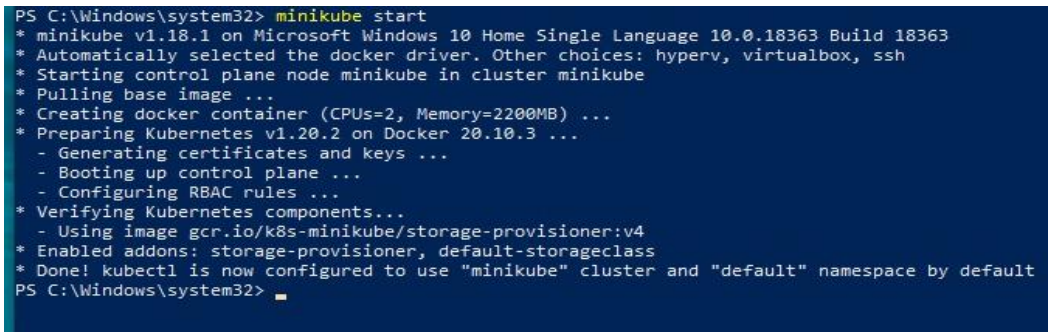
b3. Install Minikube



b4. Install Kubernetes-cli



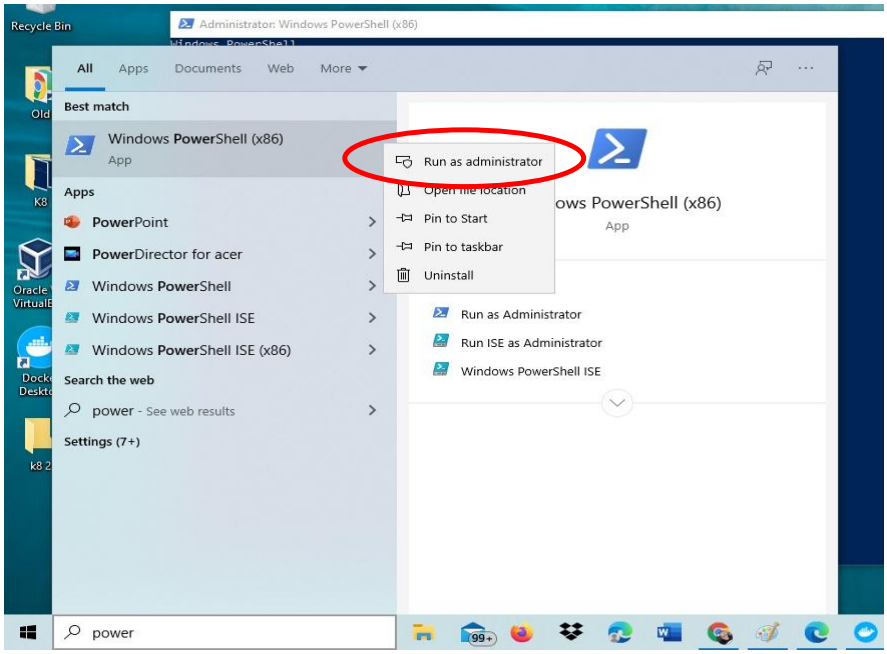
Confirm if Minikube is "working"



Now we can start developing the cluster using the installed programs

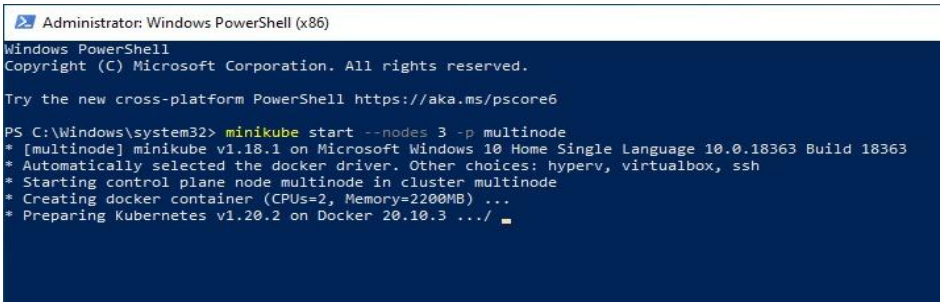
3 DEVELOPMENT PROCEDURE

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

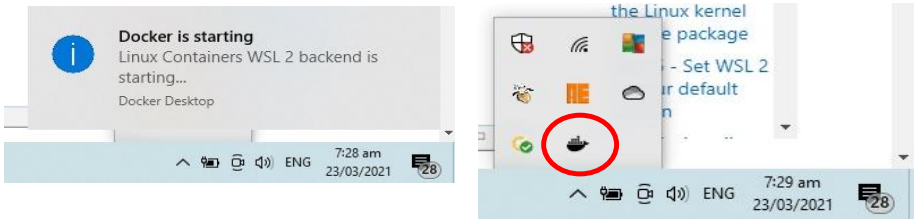


a.1 Create 3 Nodes on Minikube

CommandLine:
"Minikube start --nodes 3 -p Multinode"

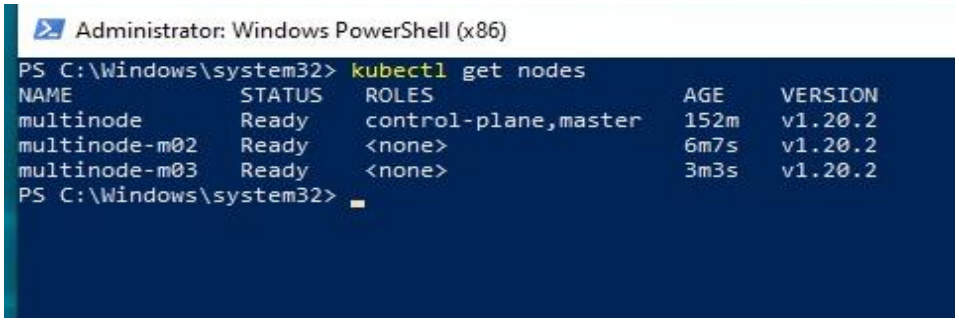


a.2 Docker must "RUN"and visible on the System Tray



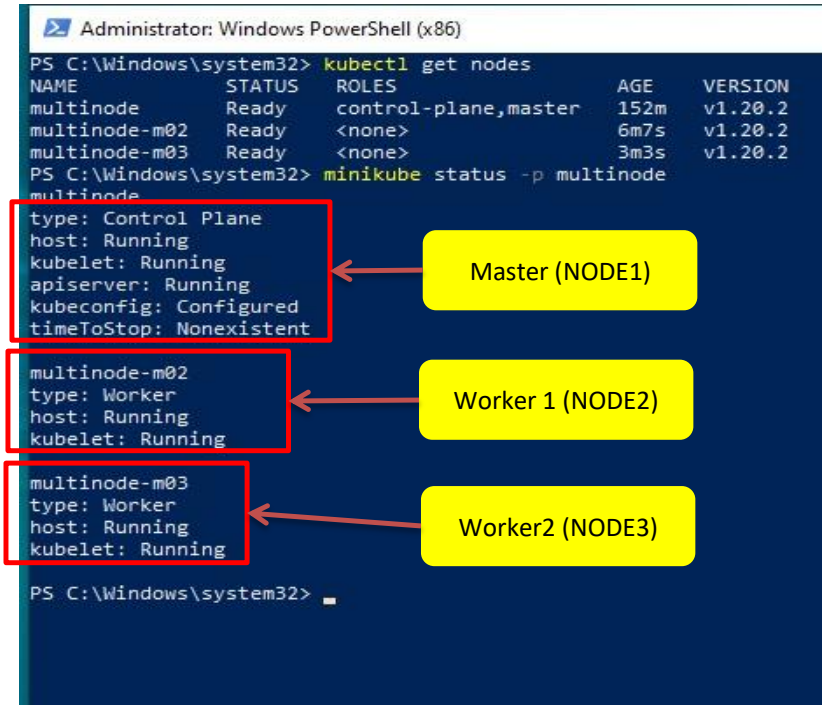
a.3 Minikube (MULTINODES) created 3 Nodes

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



a.3 Node Status

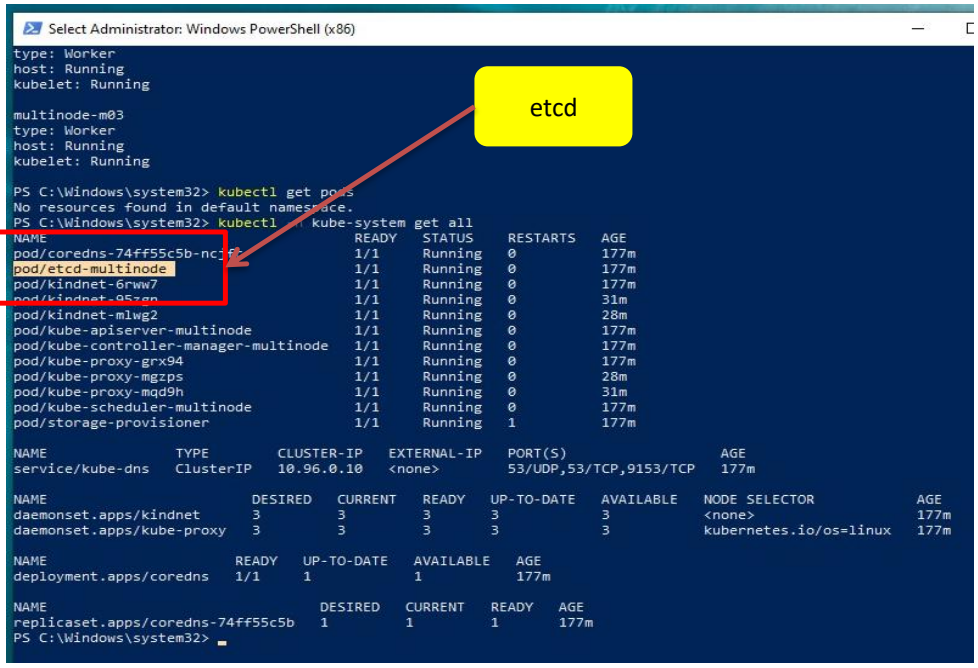
CommandLine:
"Minikube status -p Multinode"



a.4 etcd created/status

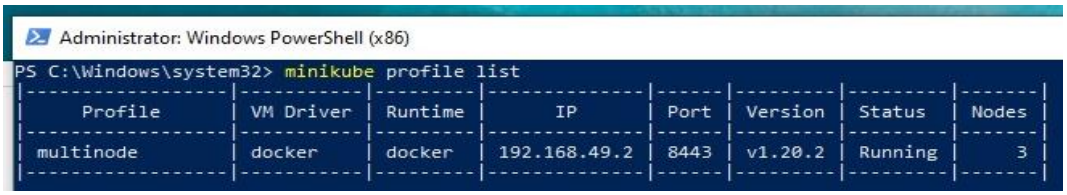
CommandLine:
"kubectl -n kube-system get all"

<pod/etcd-multinode>



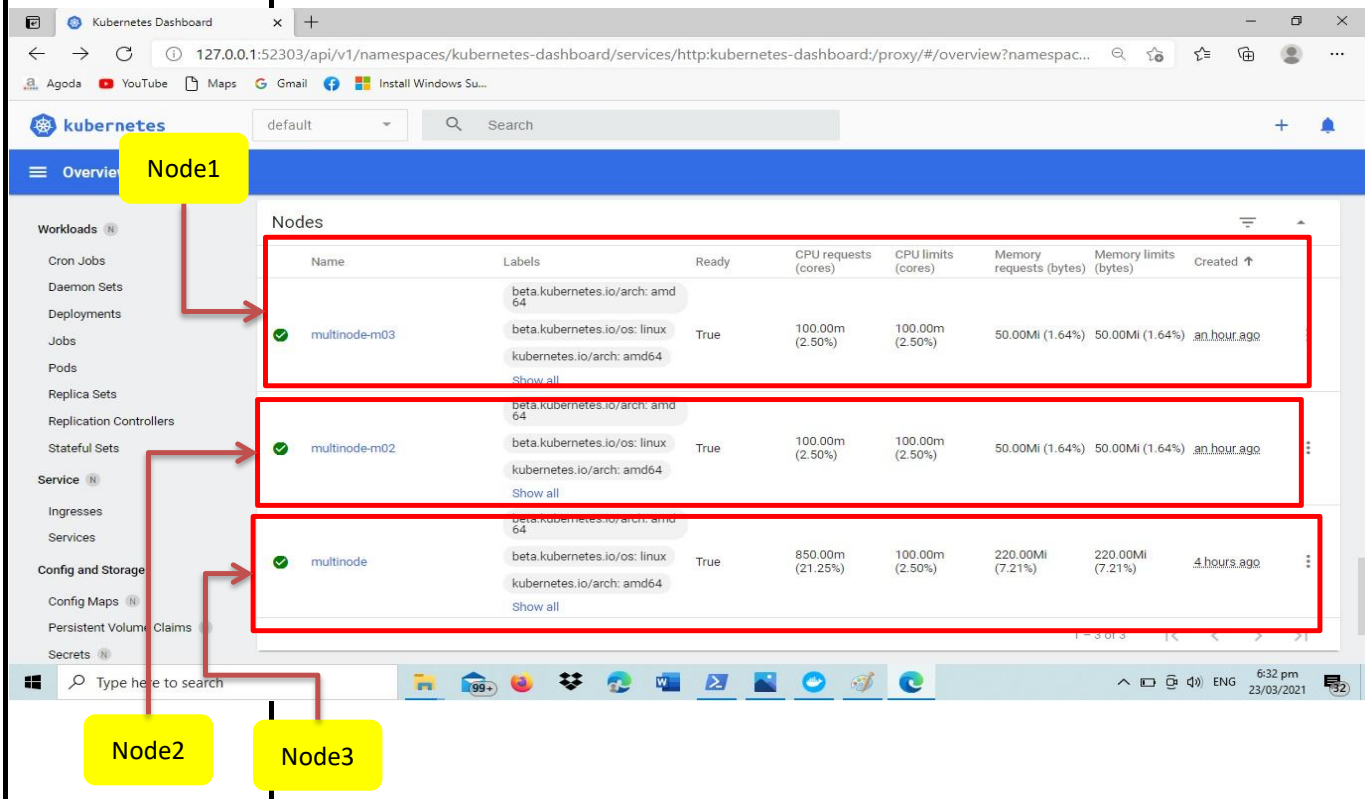
a.5 Minikube (MULTINODE) Status

CommandLine:
"minikube profile list"



a.6 Check DOCKER runs 3 Nodes

a.7 Enable Dashboard

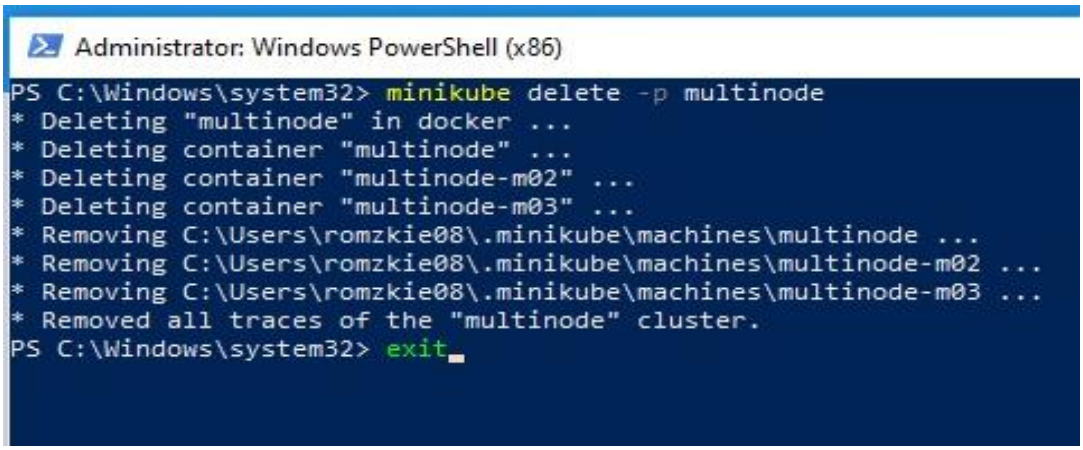


B. CLUSTER REMOVAL AND SOFTWARE EXIT

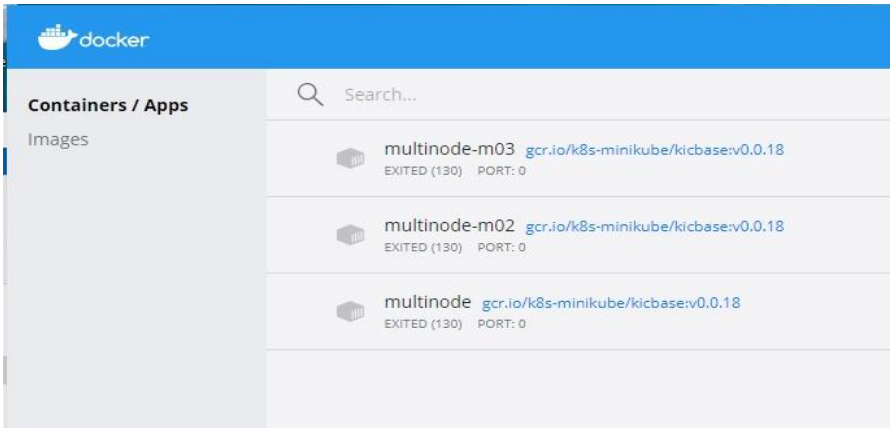
b1. Deleting CLUSTER and Container

CommandLine:

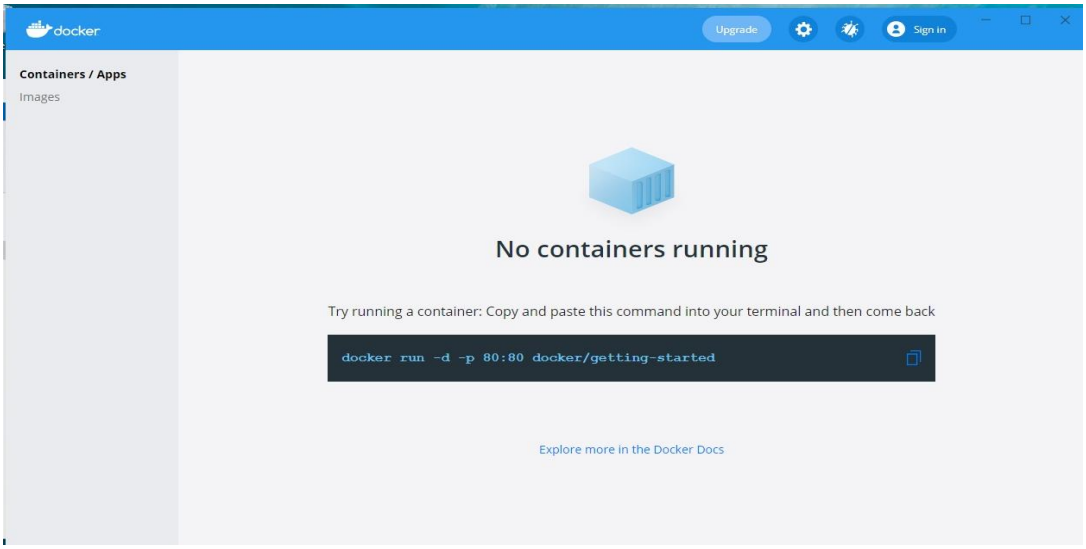
"minikube delete -p multinode"



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