Kubernetes Workshop
Software installation

If you face any issues during the installation, please contact me directly at victor.tan.33@gmail.com

Please ensure that you complete the installation **PRIOR** to the workshop, as the installation process may take over an hour to complete: this will depend on your machine specifications and broadband connection speed.

1 Install a PDF Reader

This is for viewing the lab manuals which are in PDF format. You can also view PDF files in a browser, but this can crash abruptly and disrupt your lab session. A dedicated PDF reader is more stable.

Two good options are:

https://www.adobe.com/acrobat/pdf-reader.html

https://www.foxit.com/pdf-reader/

2 Install a full-featured text editor

You will need a good text editor to open text files.

For Windows, a good option is:
Notepad++
https://notepad-plus-plus.org/downloads/v8.1.1/

For Linux/MacOS:
There should already be many preinstalled editors such as GEdit, Nano, Vim, Emacs, etc. You can also use VS Code, Atom, Brackets or any of the other popular IDEs out there.

3 Install PowerShell 7

This is only necessary if you are working on a Window system. For Linux/MacOS, you can use the standard Bash shell terminal.
NOTE: Powershell 7 is a **different** application from the standard Windows PowerShell that comes preinstalled on all Windows distribution.

The MSI installation executable can be downloaded from here:

The installation process is outlined in this Youtube video:

4 Install Docker Desktop for the Docker engine

IMPORTANT: You should have at least 8GB on your machine for Docker to run smoothly.

You will need either a WSL (Windows Subsystem for Linux) backend to provide the virtualization technology necessary to run Docker Desktop. **Using WSL2 is the fastest and most efficient way for installing Docker Desktop on a Windows system**

You can check whether WSL2 is installed on your Windows system:

If WSL is not installed, try to install the latest version (WSL2) with either of the two approaches outlined below:

https://www.omgubuntu.co.uk/how-to-install-wsl2-on-windows-10
https://gcore.com/learning/how-to-install-wsl-2-on-windows/

Detailed instructions for installing Docker on Windows

Detailed instructions for installing Docker on Linux

Some YouTube videos that demonstrate the installation process step by step
https://www.youtube.com/watch?v=ZyBBv1JmnWQ

After installation is complete, open a PowerShell terminal and type the following command

```
docker run hello-world
```

This downloads a very basic image from DockerHub, generates a container from it and produces some introductory output messages at the prompt. It validates that the installation completed successfully with the Docker engine capable of working with images and generating containers from them.

You may also need to enable BIOS-level hardware virtualization support (if it is not already enabled) in order to run a Docker engine using WSL2. If you have virtualization errors reported while installing or running Docker Desktop, then you will probably need to reboot your PC and access the BIOS menu to turn on virtualization. Some sample tutorials on doing this:

https://www.youtube.com/watch?v=ZDeje9wgDp4
https://www.youtube.com/watch?v=MOuTxfzCvMY

5 Install Kubernetes (K8s) with Minikube on Docker

With Docker Desktop already installed, the recommended option for installing Kubernetes is Minikube.

Make sure that Docker Desktop is running and active first before attempting a Minikube installation (Topic 1: Installation)

There are **2 main ways to install**:

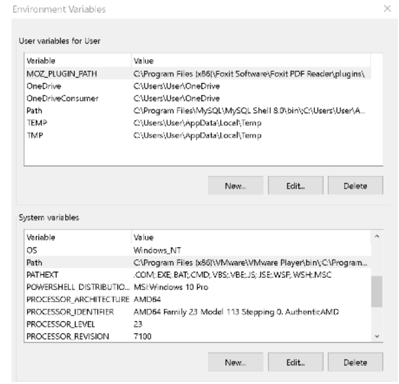
Approach 1: Select and use `.exe download` installer type (this is the default chosen and is also the easiest) and download the installer for the latest release.

Approach 2: Copy and paste the provided PowerShell commands into a PowerShell terminal
This approach is demonstrated here

The easiest is Approach 1: download and run the installer executable after it is downloaded. When installation is complete, the `minikube.exe` binary executable will be by default placed in this directory.

```
C:\Program Files\Kubernetes\Minikube
```

This directory should have been automatically added to your system PATH environment variable as part of the installation process. You can verify by checking the PATH environment variable and placing this directory value in there if it is not already automatically set.



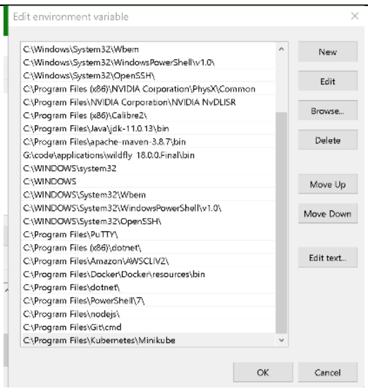
Environment Variables

User variables for User

Variable	Value
MOZ_PLUGIN_PATH	C:\Program Files (x86)\Foxit Software\Foxit PDF Reader\plugins\
OneDrive	C:\Users\User\OneDrive
OneDriveConsumer	C:\Users\User\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\Users\User\A...
TEMP	C:\Users\User\AppData\Local\Temp
TMP	C:\Users\User\AppData\Local\Temp

System variables

Variable	Value
OS	Windows_NT
Path	C:\Program Files (x86)\VMware\VMware Player\bin\;C:\Program...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
POWERSHELL_DISTRIBUTIO...	MSI:Windows 10 Pro
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	AMD64 Family 23 Model 113 Stepping 0, AuthenticAMD
PROCESSOR_LEVEL	23
PROCESSOR_REVISION	7100



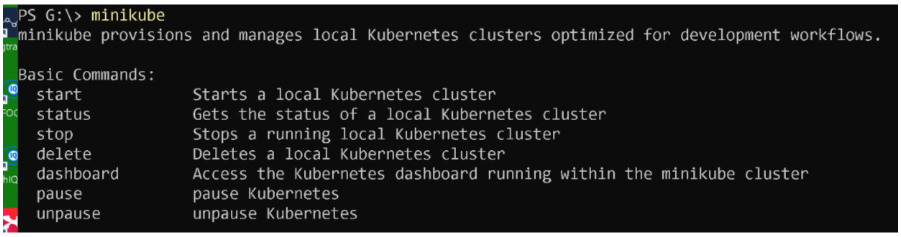
Edit environment variable

```
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Windows\System32\OpenSSH\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files (x86)\Calibre2\
C:\Program Files\Java\jdk-11.0.13\bin
C:\Program Files\apache-maven-3.8.7\bin
G:\code\applications\wildfly 18.0.0.Final\bin
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\WINDOWS\System32\OpenSSH\
C:\Program Files\PuTTY\
C:\Program Files (x86)\dotnet\
C:\Program Files\Amazon\AWSCLIV2\
C:\Program Files\Docker\Docker\resources\bin
C:\Program Files\dotnet\
C:\Program Files\PowerShell\7\
C:\Program Files\nodejs\
C:\Program Files\Git\cmd
C:\Program Files\Kubernetes\Minikube
```

Open a Powershell 7 terminal and type:

```
minikube
```

You should see some console output regarding the commands available



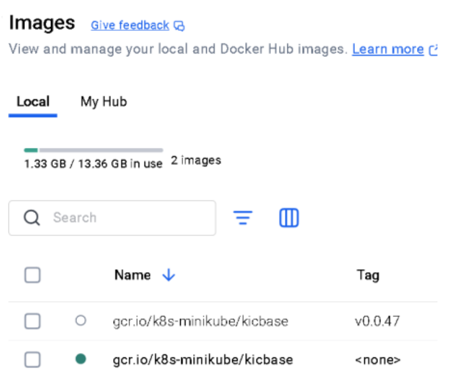
```
PS G:\> minikube
minikube provisions and manages local Kubernetes clusters optimized for development workflows.

Basic Commands:
  start            Starts a local Kubernetes cluster
  status           Gets the status of a local Kubernetes cluster
  stop             Stops a running local Kubernetes cluster
  delete           Deletes a local Kubernetes cluster
  dashboard        Access the Kubernetes dashboard running within the minikube cluster
  pause            pause Kubernetes
  unpause          unpause Kubernetes
```

We can then start Minikube cluster on the currently running Docker platform:

```
minikube start --driver=docker
```

When you start Minikube with the Docker driver, Minikube creates a Docker container that acts as a single-node Kubernetes cluster. Within this container, Kubernetes manages pods as Docker containers. You should be able to see the Minikube related images and running container from the Docker Desktop UI:

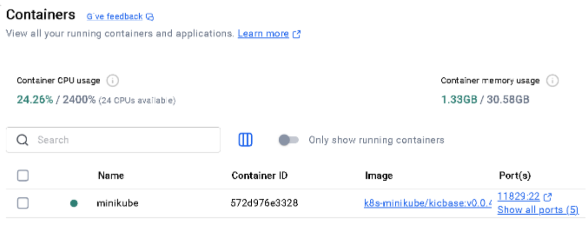


Images Give feedback

View and manage your local and Docker Hub images. Learn more

1.33 GB / 13.36 GB in use 2 images

□		Name	Tag
□	○	gcr.io/k8s-minikube/kicbase	v0.0.47
□	●	gcr.io/k8s-minikube/kicbase	<none>



Containers G've feedback

View all your running containers and applications. Learn more

Container CPU usage
24.26% / 2400% (24 CPUs availab e)

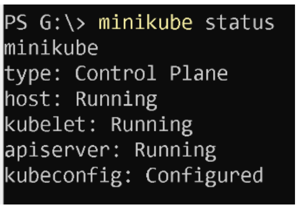
Container rnemory usage
1.33GB / 30.58GB

□		Name	Container ID	Image	Port(s)
□	●	minikube	572d976e3328	k8s-minikube/kicbase:v0.0.4	11829:22 Show all ports (5)

Next type:

```
minikube status
```

to verify that that the Minikube single node cluster is in fact up and running. You should get similar output to below.



```
PS G:\> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

If the above installation approach is not working, follow the previous Youtube video link demonstrating installation process by copying and pasting the PowerShell commands from the installation page.

After starting a newly installed Minikube, verify the K8S version that it implements with:

```
kubectl version
```

When you have confirmed that the Minikube K8s cluster is running properly on Docker Desktop, you can shut it down with:

```
minikube stop
```

and then shut down Docker Desktop

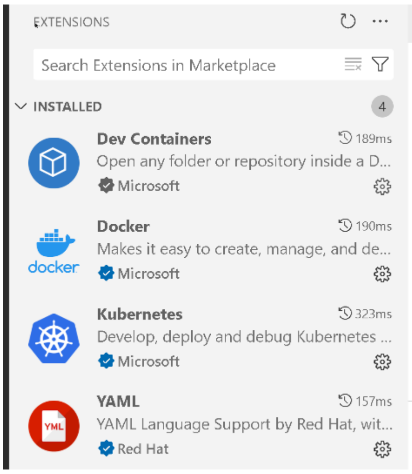
6 Install Visual Studio Code and Docker / K8s related extensions

We need an IDE to work with the source code in the sample projects.

A popular open-source IDE is Visual Studio (VS) Code
VS Code provides many useful extensions that facilitate working with Dockerfiles and YAML files (integral part of Docker).

You can install the following extensions from the marketplace for this purpose

- Docker
- Kubernetes
- Dev Containers
- YAML


EXTENSIONS

Search Extensions in Marketplace

INSTALLED 4

Dev Containers 189ms
Open any folder or repository inside a D...
Microsoft

Docker 190ms
Makes it easy to create, manage, and de...
Microsoft

Kubernetes 323ms
Develop, deploy and debug Kubernetes ...
Microsoft

YAML 157ms
YAML Language Support by Red Hat, wit...
Red Hat

If you are using your own IDE and are comfortable with that, feel free to stick with it.
All the VS code extension functionality can also be achieved from standard CLI commands.

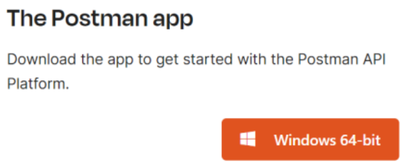
7 Install a REST API client (Postman)

We need a REST API client to send HTTP POST and GET requests to the sample apps that will be running in containers and pods. There are many REST API client options available for Windows and Linux.

A popular and widely used option among developers is Postman:

You can install it following the instructions here. Ensure you **download and install the actual app**, instead of using the browser version
https://learning.postman.com/docs/getting-started/installation/installation-and-updates/

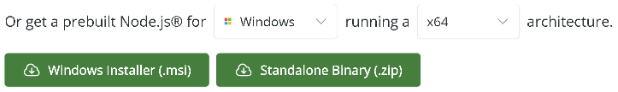
You can register to use this application for free by signing up with an account (or using your existing personal GMail account to login – via SSO).


The Postman app

Download the app to get started with the Postman API Platform.

8 Install Node.js and NPM

https://www.geeksforgeeks.org/installation-guide/install-node-js-on-windows/
Follow Method 1 above for installing Node.js from the official website
Download using Windows Installer (.msi)


Or get a prebuilt Node.js® for Windows running a x64 architecture.

Once you have finished installation, type the following 2 commands in a command prompt in Windows to validate successful installation of Node and NPM.

```
node -v

npm -v
```

9 Sign up for a DockerHub account

If you don’t already have one, sign up for a free DockerHub account at:

https://hub.docker.com/

Instructions are outlined here
https://docs.docker.com/docker-id/

10 Preparations

10.1 Broadband connection bandwidth

A significant portion of lab activity involves downloading images from the DockerHub public registry. Some of these images may be up to 20MB or more in size. If your broadband connection has low bandwidth, these downloads may take excessively long to complete and delay your progress in the lab. If possible, try to ensure a connection speed of at least 5Mb/s during the Docker lab sessions.

10.2 Basic Linux CLI and Docker knowledge

K8s is a container orchestration platform for Docker (and other similar container platform) containers. Some basic knowledge of Docker would be very useful.

Basic Docker tutorials:
https://www.datacamp.com/tutorial/docker-tutorial
https://www.hostinger.com/tutorials/docker-tutorial

You will need some knowledge of basic Linux shell commands to work inside Docker Linux containers. If you have no experience at the command line in Linux, you can optionally prepare beforehand as we will not have time to do a beginner level Linux tutorial in the course.

Basic Linux tutorials:
https://ubuntu.com/tutorials/command-line-for-beginners
https://ryanstutorials.net/linuxtutorial/
http://www.ee.surrey.ac.uk/Teaching/Unix/