**Installing Podman/Minikube/CRI-O as Docker Desktop alternative**

You are a few steps far from moving away from Docker Desktop to a much powerful free alternative, which will enable you to introduce Kubernetes in your development environment if desired, or continue working as normal with plain Docker-like containers.

**The given commands are expected to be run in a PowerShell console.  
For the moment it is impossible to use minikube with the VPN**

**Minimum requirements: disk: 40GB  
 memory: 3GB**

**1. As administrator:**

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

Reboot if asked to do so.

([adsi]"WinNT://./Hyper-V Administrators,group").Add("WinNT://$env:UserDomain/$env:Username,user")

logoff;exit

You will be logged off. This is needed to make Windows aware of the needed group membership. Do not skip this step.

Install:<https://storage.googleapis.com/minikube/releases/latest/minikube-installer.exe>

**2. As standard user:**

Then execute the script “installation\_podman.ps1”  
For that, you need to open a new powershell console, type:  
cd <path\_to\_the\_extracted\_archive>  
.\installation\_podman.ps1

Now you can use podman <command> (or its alias docker <command>) and minikube kubectl — <command> freely

**How to use podman and the most used commands**The goal of this part is to give you the basics of minikube and podman, some links are given at the end if you want to go further with podman

**3. Minikube**

"Minikube is a tooll that let you run Kubernets locally. It runs a single-node Kubernetes cluster on your computer"

Podman is normally not compatible with Windows, but with minikube, we will have a linux Virtual Machine with podman installed, and on your windows will be installed a podman remote client. Everytime you use podman on windows it will connect to the VM and execute the podman command.

**Note**: Be careful when you try to acces an application in a container (for example accesing with your web navigator to a page that your app generated. With Docker desktop you had tu connect to localhost, with podman, you will need to connect to the IP of the VM, to get the IP, the powershell command is:

**minikube ip**

You will also need these commands:

**minikube start** (everytime you start your computer, minikube is shut down, you will need to start it before using podman)

**minikube stop** (to stop minikube correctly)

**minikube pause** (put kubernetes in pause but the VM will still be running, so if you just need podman we advice you to use this command to get more performance)

**minikube delete** (it will delete your minikube node, see next command if you want to create a new one)

**minikube start --container-runtime=cri-o --cpus 4 --memory $(((Get-CimInstance Win32\_PhysicalMemory | Measure-Object -Property capacity -Sum).sum - 6gb) /1mb)**(to create a new node, only works if there is not an existed node)

you can also check **minikube --help** if you want to know more commands

For information, everytime you stop/start your VM (with minikube stop/start), nothing will be saved. You should not modify anything directly on the VM.

Sometimes, If you shut down your VM with container still existing (even not running), when you start again your VM you may have problems with name of container already in use but that same container name does not exist if you want to erase it.

**4. How to use Podman**

With the installation of Podman an alias for docker has been created, so when you write docker it will use podman.

The vast majority of commands that you were using with docker will work exactly the same with podman, but note that docker compose won't work (see part 5), and there's nothing compared to swarm.

Podman will read dockerfile the same way Docker Desktop was doing it.

For those who haven't use docker here is a quick memo of basic commands:  
the example comes from the docker "getting started" class on pluralsight, here is the link if you need to check: (Do not hesitate to check it, as we won't go into details here)  
<https://www.pluralsight.com/courses/getting-started-docker>

A: build an image  
You will need a Dockerfile, example:  
This file is the instructions to build your app

Once you got your Dockerfile, with powershell place your prompt in the folder of your application, and execute:

**"podman image build -t <tag> ."**, if you want to push it on an online repository, the tag should look like that: <user>/<repository>:<tag>

Then you can push it in a registry (optional):

**"podman image push ultimatom/test1:first\_ctr --creds <user>:<password>"** automatically it uses the docker hub registry.  
there should be an action to store automatically the credentials, to check

then you can run it:

**podman container run --name <name\_of\_your\_container> <tag>** You may need to define mapping for ports, example:

**podman container run --name test1 -p 8000:8080 ultimatom/test1:podman\_first\_test** with the p flag, when I want to contact the port 8080 of my container, I will need to contact the port 8000 of the VM

then you can check your container running

**podman container ls**

You will need to stop/start or delete your container sometimes

**podman container stop <name\_of\_your\_container>  
podman container start <name\_of\_your\_container>  
podman container rm <name\_of\_your\_container>**

**5. podman-compose**

podman does not include function to launch interdependent containers.

for that a python script exists. This script can read docker-compose.yml file, will create a pod then launch your different container inside the pod.

What is a pod ?

"Pods are a way of grouping containers together inside their own namespace, network, and security context"  
For more informations about a pod: <https://developers.redhat.com/blog/2019/01/15/podman-managing-containers-pods#:~:text=The%20Pod%20concept%20was%20introduced,other%20containers%20to%20the%20pod>.

At this moment, it is not possible to use podman-compose in windows, and it is not included in minikube yet.

So we made a script to make it work.  
For information: this script is creating a unix container with podman-compose installed on it and will use this container to generate all the command that podman-compose will execute then execute them inside the minikube VM.

To call this script, you just need to be in the directory containing your docker-compose.yml. And type:

**podman-compose**  (this is an alias for: C:\Users\<user>\Downloads\podman-2.2.1\podman\_compose\_Windows\_part.ps1)

there is two flag that can be used, "**-m**" and "**-u**"

- it needs to have a share folder between the windows host and the minikube VM, to create it you need to add the "**-m**" flag, this will open a new powershell prompt in background, wait 30 sec so that the sharing is effective, then the script will keep going.  
after the end of the script the prompt windows for the folder sharing will still be up, **we advise you to let it run, if you need again to use the scrip you will win 30 sec because you won't need that flag.**

- It is possible that the folder inside the VM that the folder created inside the VM by the sharing process becomes corrupted, in this case, close the powershell prompt running the sharing process if it has not been done yet, and use the "**-u**" flag, this flag will unmount the folder inside the VM then delete it, if you activate this flag it will automatically activate the "**-m**"flag, so it won't be necessary with this one

**6. Uninstall Podman**

Inside the folder installed by podman (in download you have a folder podman-2.2.1)

There is a powershell script to uninstall podman if needed.

**7. More documentation about Podman**

Minikube documentation:  
<https://minikube.sigs.k8s.io/docs/start/>

Oficial documentation of Podman  
<https://docs.podman.io/en/latest/>

News feed about podman:  
<https://podman.io/>

Github for podman and podman-compose:  
<https://github.com/containers/podman>  
<https://github.com/containers/podman-compose>