# Solution M3: Containerization

There are multiple ways to achieve what was requested

As the visual approach is intuitive, the presented solution is implemented with PowerShell

Please note that the solution may be far from being optimal. The aim is to illustrate all (or most) individual steps

## Possible Solution

Check the **M3-Solution-Script-Containerization.ps1** file

Adjust it to match your situation

Open a **PowerShell** session with **Run as Administrator** on **the Hyper-V** host and execute it

Here, we will explain just the steps related to the actual solution

Once the domain is up and running and all machines are part of it, we can continue

Role installation. We can go only with **Containers** role

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { Install-WindowsFeature -Name Containers -IncludeManagementTools -Restart }**

Install Docker Provider

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force ; Install-Module -Name DockerMsftProvider -Repository PSGallery -Force }**

Install Docker

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { Install-Package -Name Docker -ProviderName DockerMsftProvider -Force }**

Ensure the Docker service is started

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { Start-Service docker }**

Check that the Docker is working and reachable

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker version }**

Pull the image

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker image pull mcr.microsoft.com/windows/servercore/iis }**

Start a container

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker container run -d --name web1 -p 9000:80 mcr.microsoft.com/windows/servercore/iis }**

Check if the container is running

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker container ls }**

And if yes, try to access it

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { curl http://localhost:9000 -UseBasicParsing }**

Create a folder to accommodate the custom image creation process

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { New-Item -Type Directory -Path C:\Temp }**

Download the supporting files

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { curl -UseBasicParsing https://zahariev.pro/files/web.zip -OutFile C:\Temp\web.zip}**

Expand the compressed archive

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { Expand-Archive C:\Temp\web.zip -DestinationPath C:\Temp }**

Build the image

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker image build -t hw-iis C:\Temp }**

Check if the image has been created

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker image ls }**

Start a container out of our new image

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker container run -d --name web2 -p 8080:80 hw-iis }**

Check if the container is running

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { docker container ls }**

And if yes, try to access it

**Invoke-Command -VMName $VM2 -Credential $DC -ScriptBlock { curl http://localhost:8080 -UseBasicParsing }**