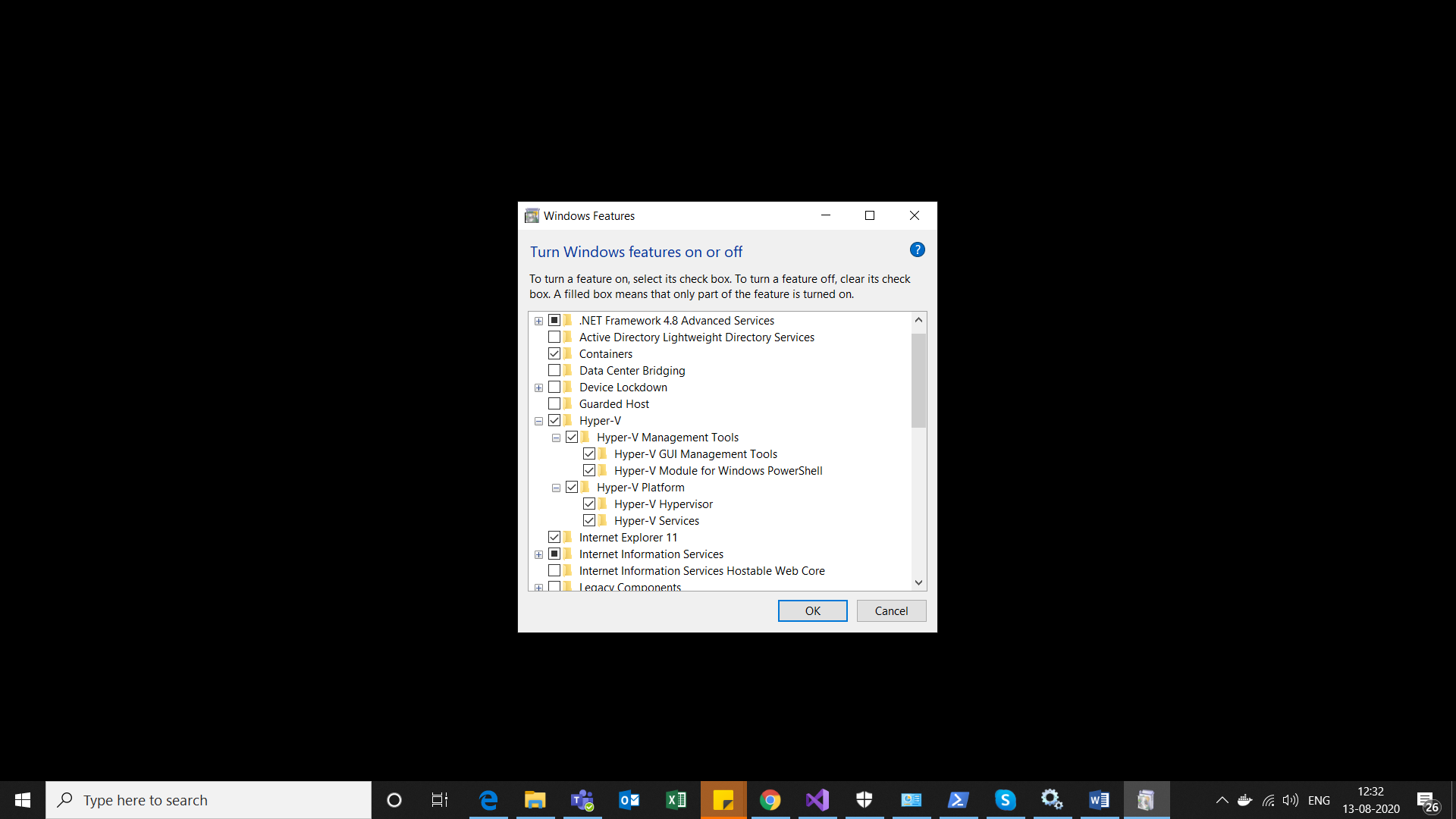
# Check this before installation

## Windows 10 64-bit: Pro, Enterprise, or Education (Build 15063 or later).

## Hyper - V and Containers should be enabled

* Navigate to Turn Windows Feature on/off on your machine and enable Hyper - V and Containers



* You can also run below commands in PowerShell or command prompt to enable Hyper - V and Containers(Run by administrator)

1. Enable-WindowsOptionalFeature -Online -FeatureName containers –All

(Restart your machine after running this command)

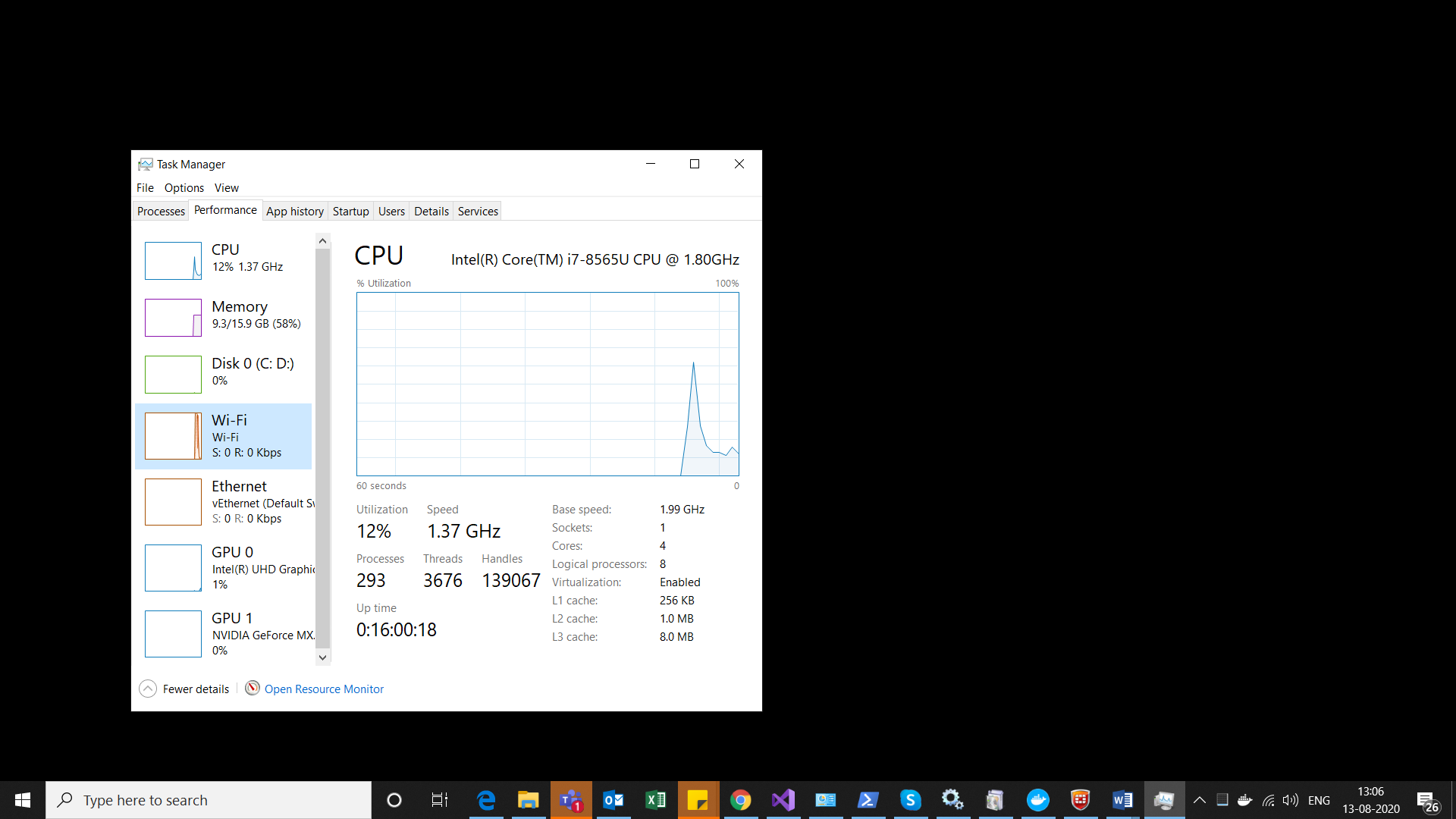
1. Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V –All

(Restart your machine after running this command)

Reference link : <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/quick-start/enable-hyper-v>

## Virtualization is enabled

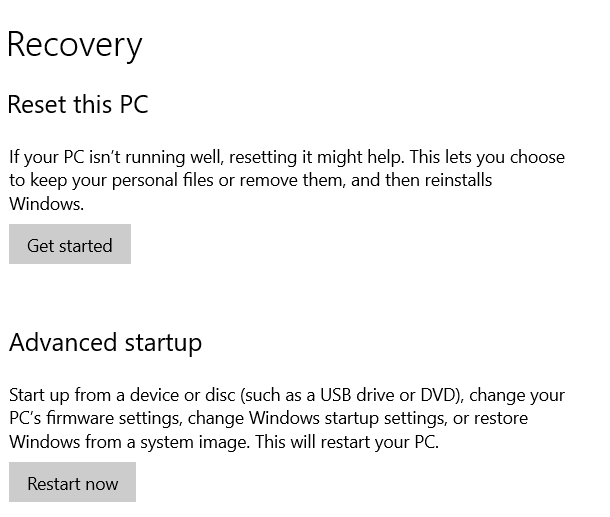
To check if virtualization is enabled, open Task Manager, navigate to Performance tab



If the Virtualization is disabled, then you need to enable the Virtualization via BIOS settings.

Please refer to below steps to enable Virtualization

* Navigate -> Change advanced startup option -> Restart now



* Select UEFI Firmware settings
* Bios - Enable Intel(R) Virtualization Technology

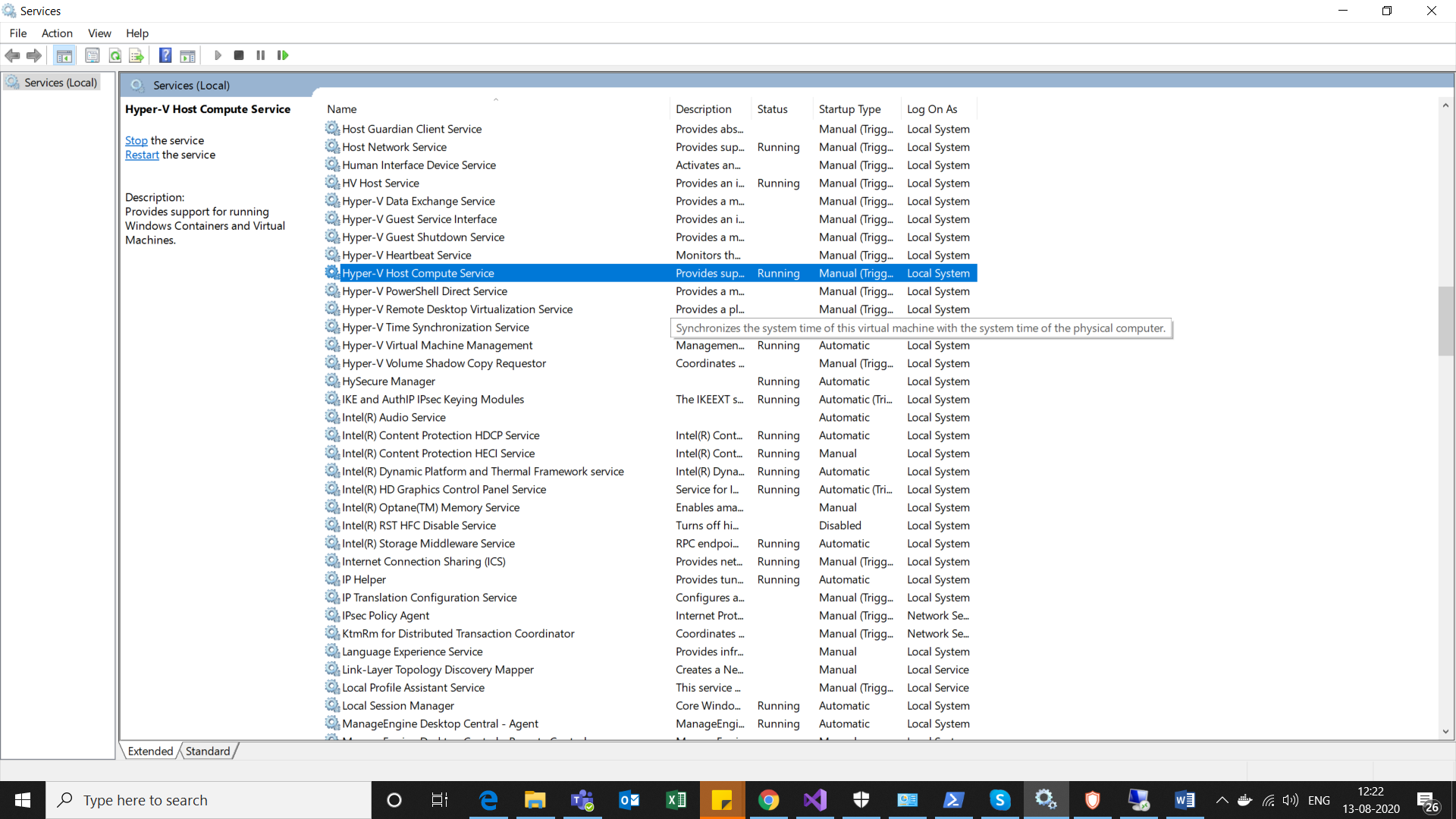
These settings can be different from machine to machine. Please refer to below link for more details.

<https://2nwiki.2n.cz/pages/viewpage.action?pageId=75202968#:~:text=ON%20the%20System.-,Press%20F2%20key%20at%20startup%20BIOS%20Setup.,changes%20and%20Reboot%20into%20Windows.>

## Check two services on your machine

* The following two services should be running on your machine.

1. Hyper-V Host Compute Service
2. Hyper-V Virtual Machine Management



* If you don’t see Hyper - V Host Compute Service, please follow below instructions

1. Open "Window Security"  
2. Open "App & Browser control"  
3. Click "Exploit protection settings" at the bottom  
4. Switch to "Program settings" tab  
5. Locate "C:\WINDOWS\System32\vmcompute.exe" in the list and expand it  
6. Click "Edit"  
7. Scroll down to "Code flow guard (CFG)" and uncheck "Override system settings"  
8.Start vmcompute from powershell "net start vmcompute"

Reference link : <https://github.com/docker/for-win/issues/3597>

# Docker Desktop installation

* Get the docker desktop for Windows (stable version)from this URL : <https://hub.docker.com/editions/community/docker-ce-desktop-windows/>
* Install by running the exe (Run by administrator)
* Enable Hyper-V Windows Features when prompted on the Wizard

# Start Docker

* To start docker, search for Docker Desktop on your machine and double click on it
* A whale icon appears in the notification bar, on hover it shows ‘Docker is starting’

# Running Docker

* When the whale icon is steady, on hover it shows ‘Docker is running’
* By default, linux containers are started. You should use linux containers only.
* In services, Docker Desktop service is running.

# About Docker Desktop

* Click on the whale icon, click on About Docker Desktop
* Docker desktop includes **Docker Engine, Docker compose, Kubernetes, Notary and Credentials Helper is included**

# Basic Docker commands

Command prompt or PowerShell (Run by administrator)

* Check Docker version : docker -v
* Check list of all images : docker images
* Check list of containers currently running : docker container ls
* Check list of all containers : docker container ls -a
* Build images : docker build -t <<imagename>> -f ‘location of docker file’ <<location of the sln>> (imagename has to be lowercase)
* Run images : docker run <<imagename>> (imageId can also be used)
* Remove images : docker image rm <<imagename>>

# Docker build and run images

## Example 1

Project : VictimNotification \ ScramNet.Ally.AssignVictimClient.

Docker support is already added for this project. Dockerfile is at solution level.

### Build images

If your code uses environment variables, you will need a DockerDevfile for building docker images locally. DockerDevfile is same as Dockerfile only difference is that in the DockerDevfile add the environment variables needed to run the project. Place these env variables just before the entrypoint. Place the DockerDevfile at the same level as the Dockerfile.

ENV ConfigName1="ConfigValue"

ENV ConfigName2="ConfigValue"

ENV ConfigName3="ConfigValue"

Note : There should not be any space before or after =

For visual studio build, these environment variables can be read from the launchSettings.json

Command : docker build -t scramnetgpsallyassignvictimclientenv -f 'D:\Source\Repos\ScramNet.Ally.AssignVictimClient\DockerDevfile' --build-arg PAT= ENTER YOUR PAT HERE D:\Source\Repos\ScramNet.Ally.AssignVictimClient



If you are at solution level in the command prompt or powershell, then you can use following command

Command : docker build -t scramnetgpsallyassignvictimclient -f ./DockerDevfile --build-arg PAT= ENTER YOUR PAT HERE .

Note : There is a space and dot after the PAT token.

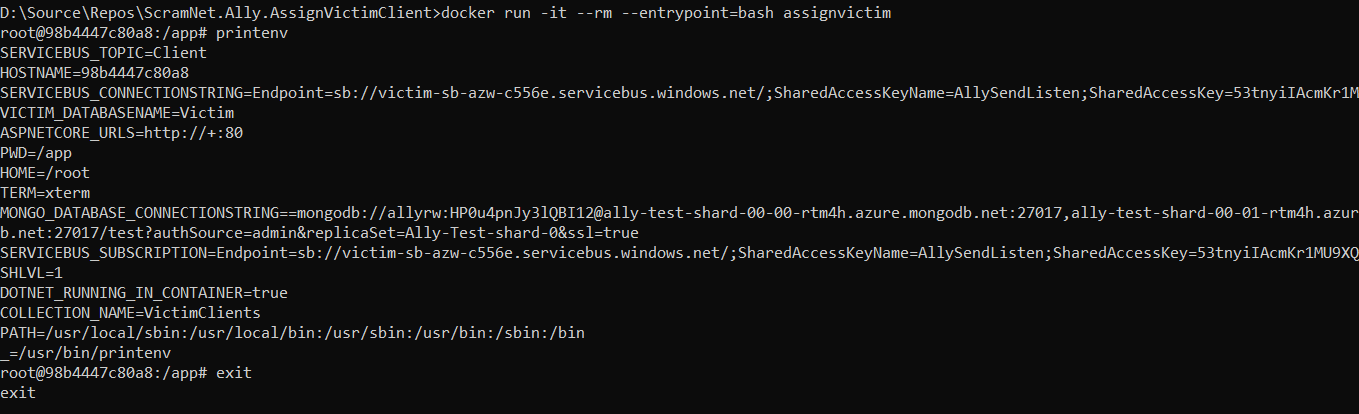


* -t : name of the image
* -f : path of the docker dev file
* --build-arg PAT : we need to pass PAT token so as to authenticate the container to access the AMS Nuget feed
* Path to the sln file

Once the image is built, we can check the values for the environment variables. Please run following commands.

Command:

1. docker run -it --rm --entrypoint=bash scramnetgpsallyassignvictimclientenv
2. printenv
3. exit



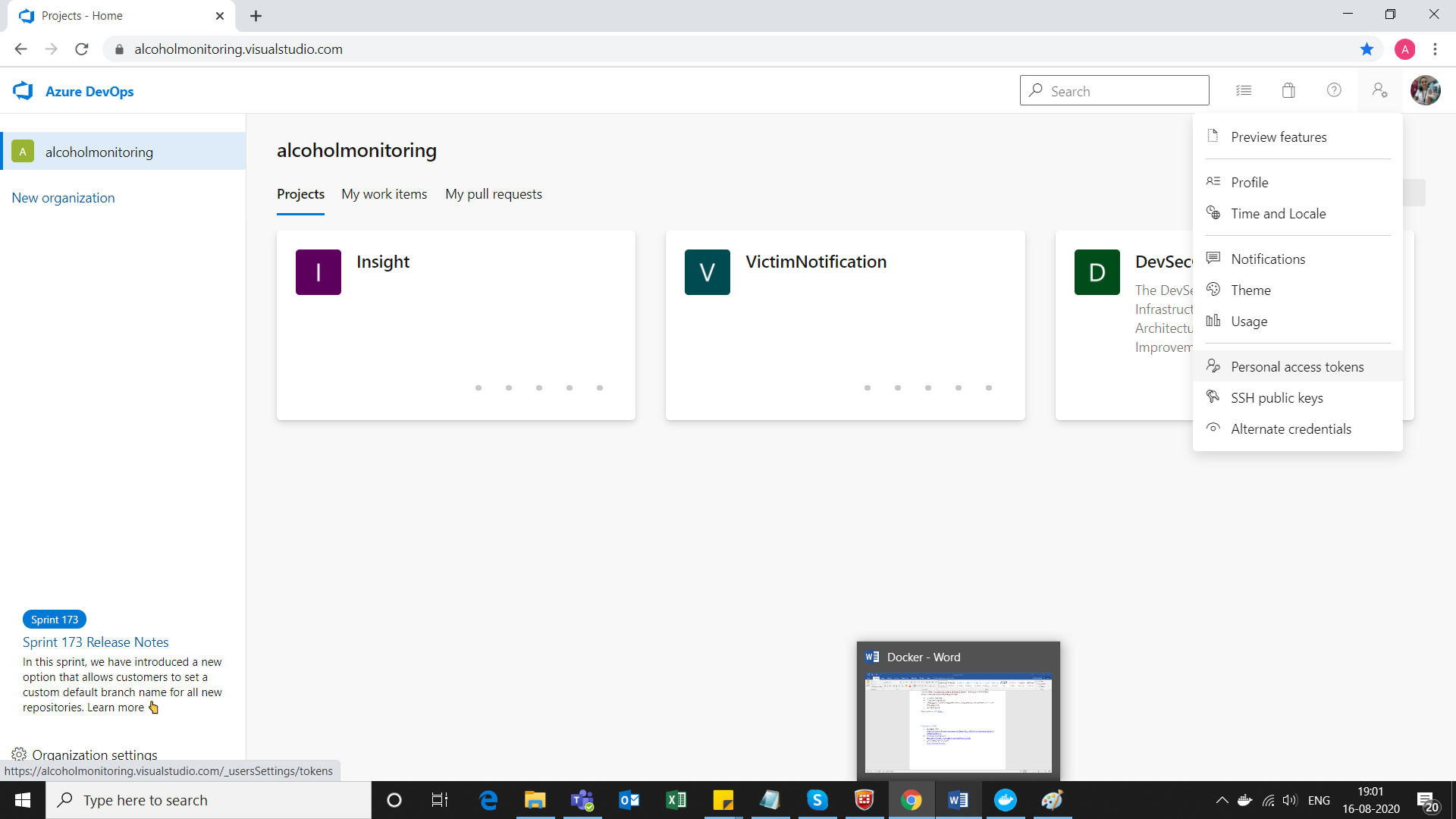
### Run Images

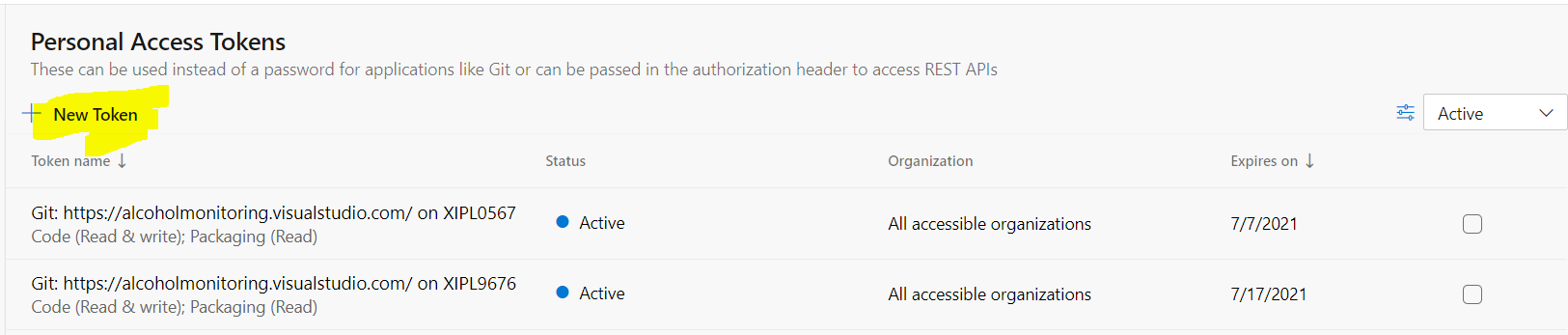
Command : docker run -it --rm scramnetgpsallyassignvictimclientenv

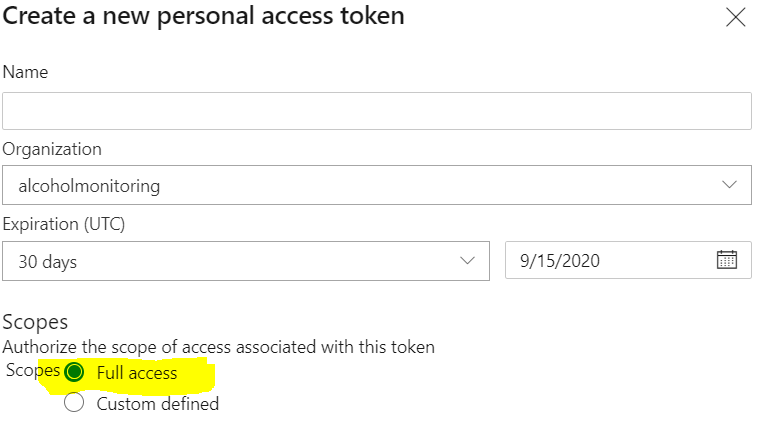
#### How to generate PAT token ?

User Settings -> Personal Access token -> New token -> Full Access

Make sure to copy and save this token locally.







Reference link : <https://alcoholmonitoring.visualstudio.com/DevSecOps/_wiki/wikis/Infrastructure.wiki/231/Docker-Deployments>

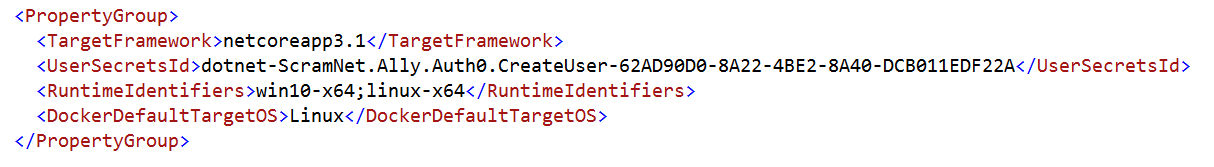
## Example 2

Project VictimNotification \ ScramNet.Ally.Auth0.CreateUser.

Docker file is at project level.

In the csproj file, add the following

Add <RuntimeIdentifiers>win10-x64;linux-x64</RuntimeIdentifiers>



This will provide option of linux target runtime during Publish

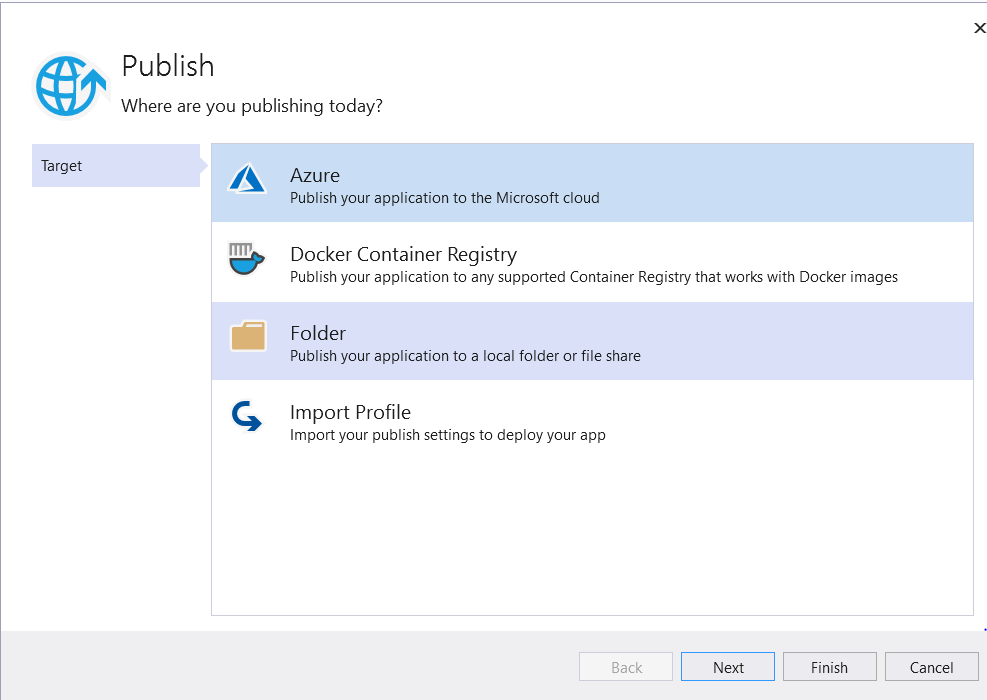
### Publish

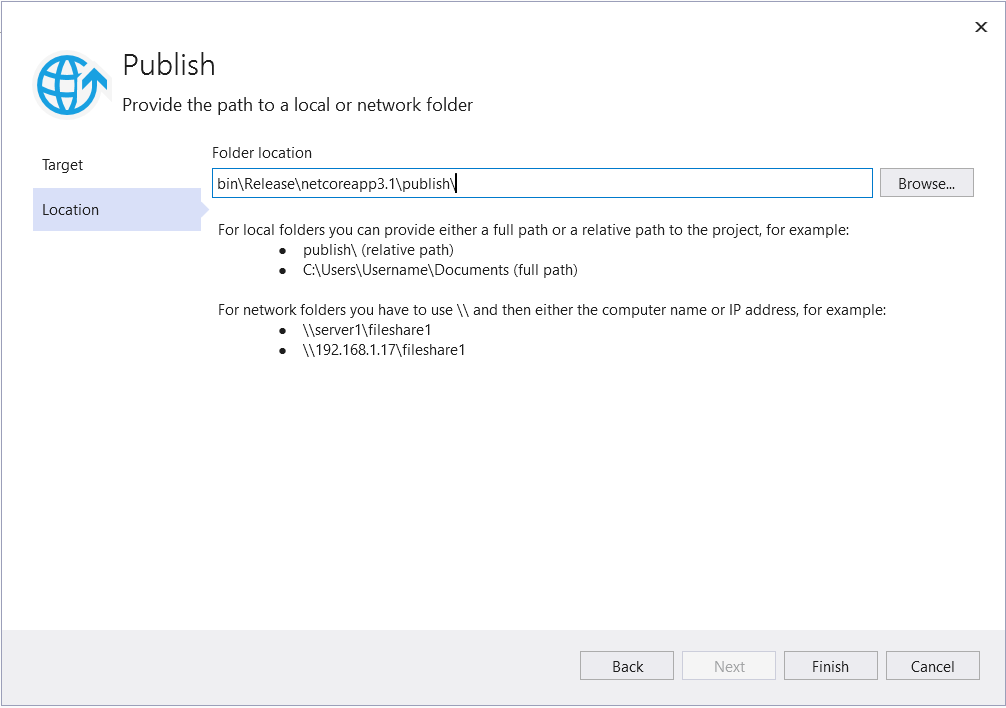
### Right click on project ->Publish -> Publish as a Folder -> Finish

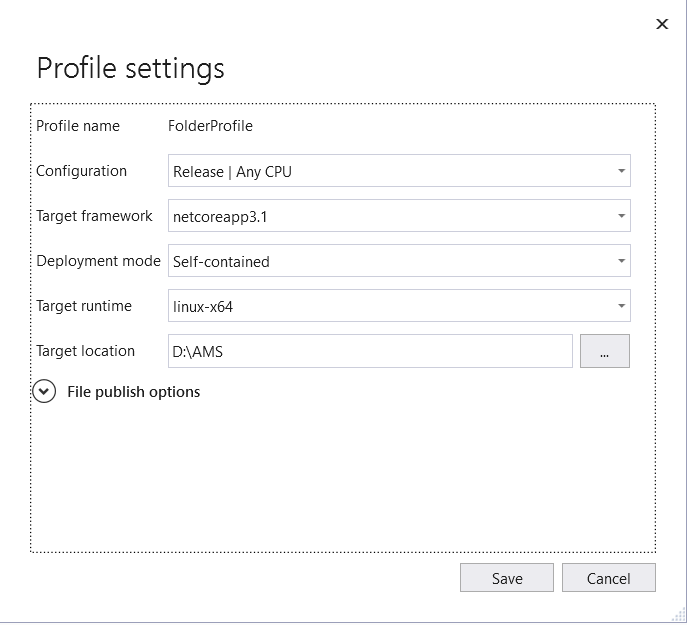
Edit Configuration -> Release | Any CPU

Deployment mode -> Self contained

Target Framework -> linux-x64







### Build Images

Open Command Prompt or Powershell (Run as administrator)

Change directory to the path where the App is published.

In this case, it is D:\AMS

Command : docker build -t scramnetallyauth0createuser .

Note : After imagename there is a space and then a dot.

### Run Images

Once the image is built, you can run it using the following command

Command : docker run -it --rm -p 8080:80 –e ASPNETCORE\_ENVIRONMENT=value –name <<containername>> <<imagename>>

docker run -it --rm -p 8080:80 -e ASPNETCORE\_ENVIRONMENT=stage --name scramnetallyauth0createusercontainer scramnetallyauth0createuser

* -it : This means interactive. We can see the logs on the command prompt when the app is running
* --rm : This means remove the container once it is exited
* -p : In this case it will map local host 8080 to container port 80
* -e : To specify the environment value
* --name : To specify the name of the container

Reference link : This the document Sapandeep has created on the same topic

<http://sharepoint.ams.local/Technology/NetDev/SCRAMnetGPS/Shared%20Documents/docker_run.docx?Web=1>

# Docker Pull Images

Install Azure CLI using this link :

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-windows?view=azure-cli-latest&tabs=azure-cli>

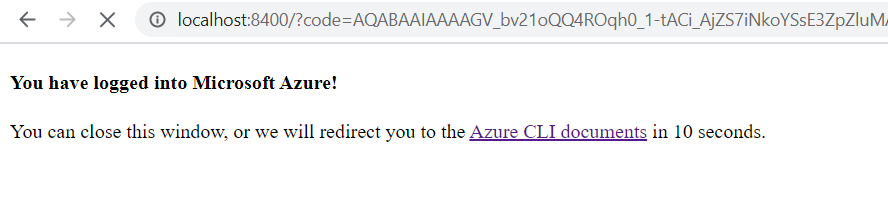
Run the downloaded exe.

Open powershell or command prompt (Run as administrator)

Command : az login

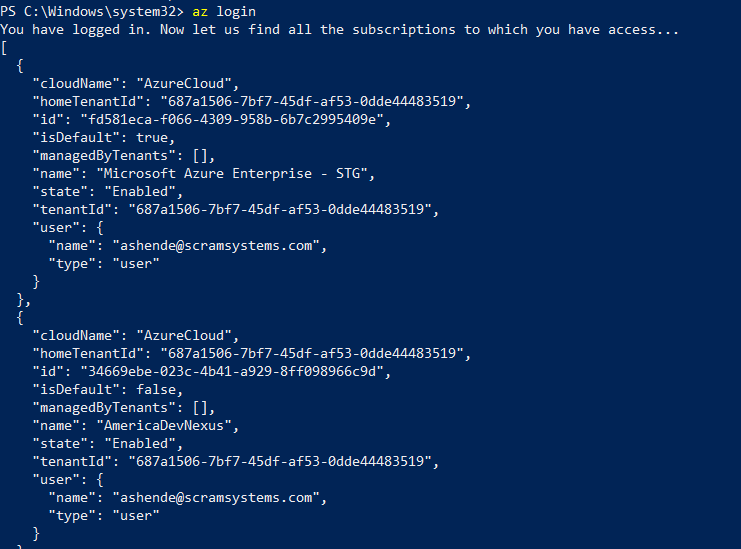
This will open a browser, when prompted select your scram account and provide password if required.

You will see this message on the browser



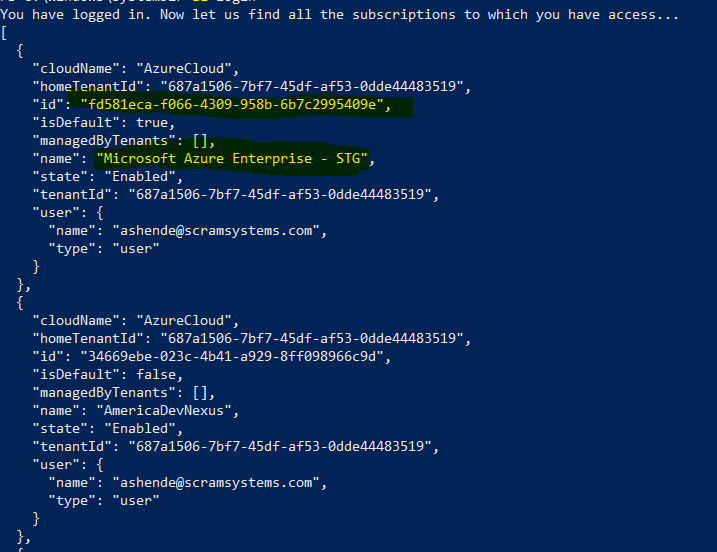
Open the same powershell or command prompt window, it shows logged in

Here we see the list of all subscriptions accessible to us.



Command : az account set --subscription={{Microsoft Azure Enterprise – STG}}

Note : You get this subscription id from the list of all accessible subscriptions



Command : az acr login -n scramnetwork

Run above command and then we can start pulling images using below command

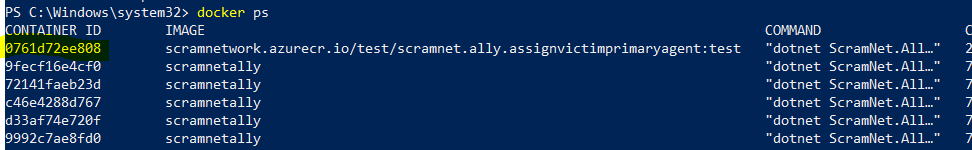
Command : docker pull <<imagename>>

docker pull scramnetwork.azurecr.io/test/scramnet.ally.assignvictimprimaryagent:test

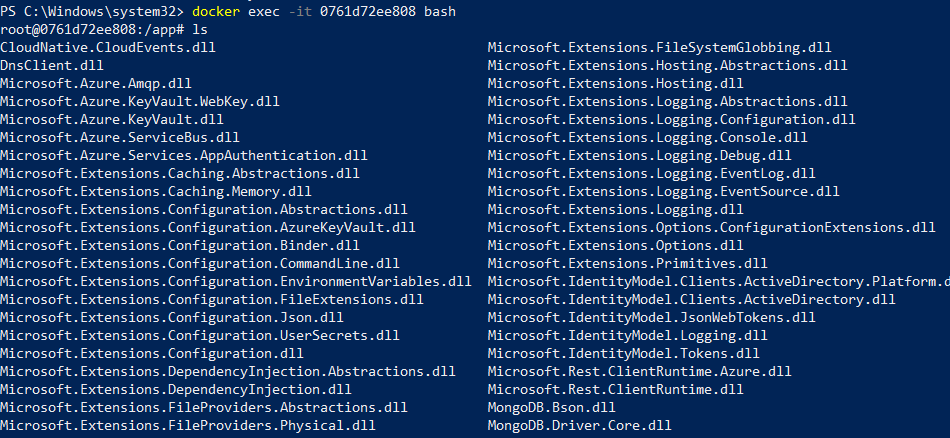
Command : docker run --rm scramnetwork.azurecr.io/test/scramnet.ally.assignvictimprimaryagent:test

Now, when we list the containers we can see a new container for the pulled image

Command : docker ps



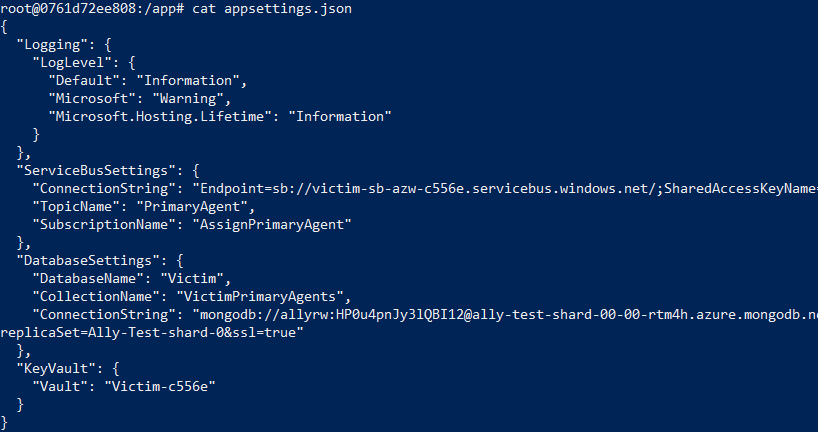
Copy the container id and run following command

Command : docker exec -it <<containerId>> bash

To see the contents of an individual file, use following command

Command : cat <<filename>>

Note : We have not exited from the previous command

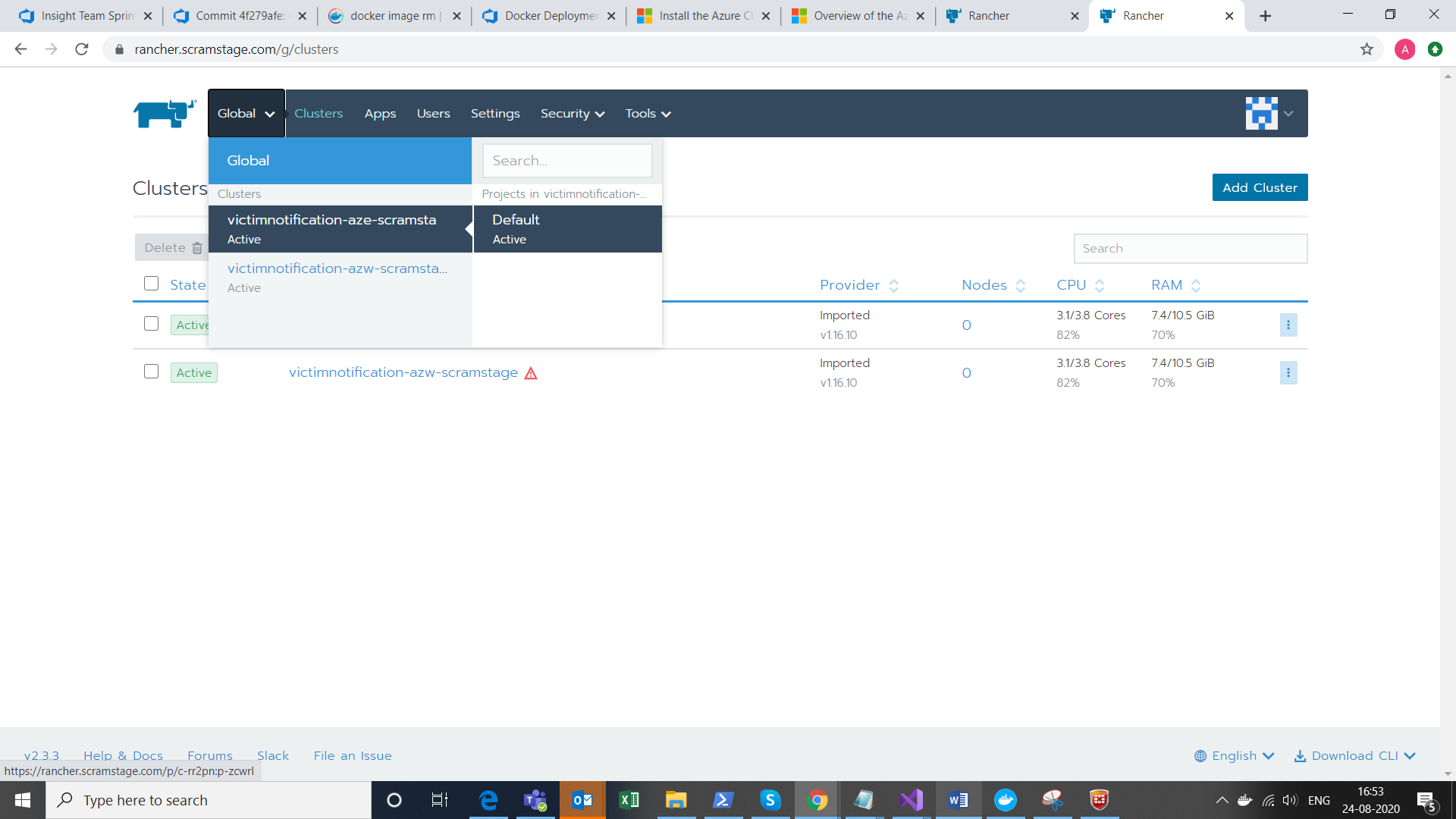


#### How to get the imagename ?

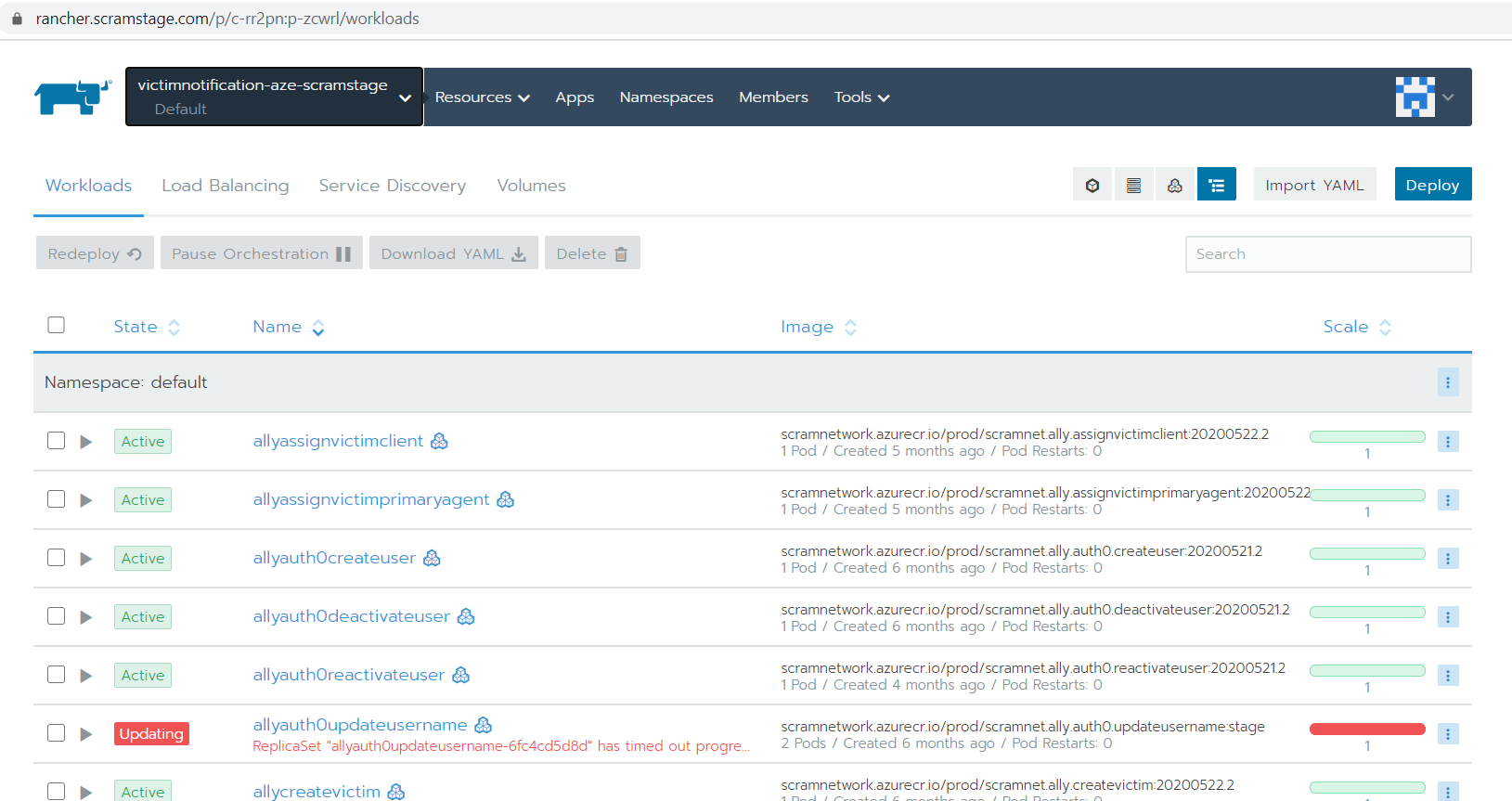
Login to rancher using this link : <https://rancher.scramstage.com/g/clusters>

(This is for stage env)

Navigate to Default(Active)

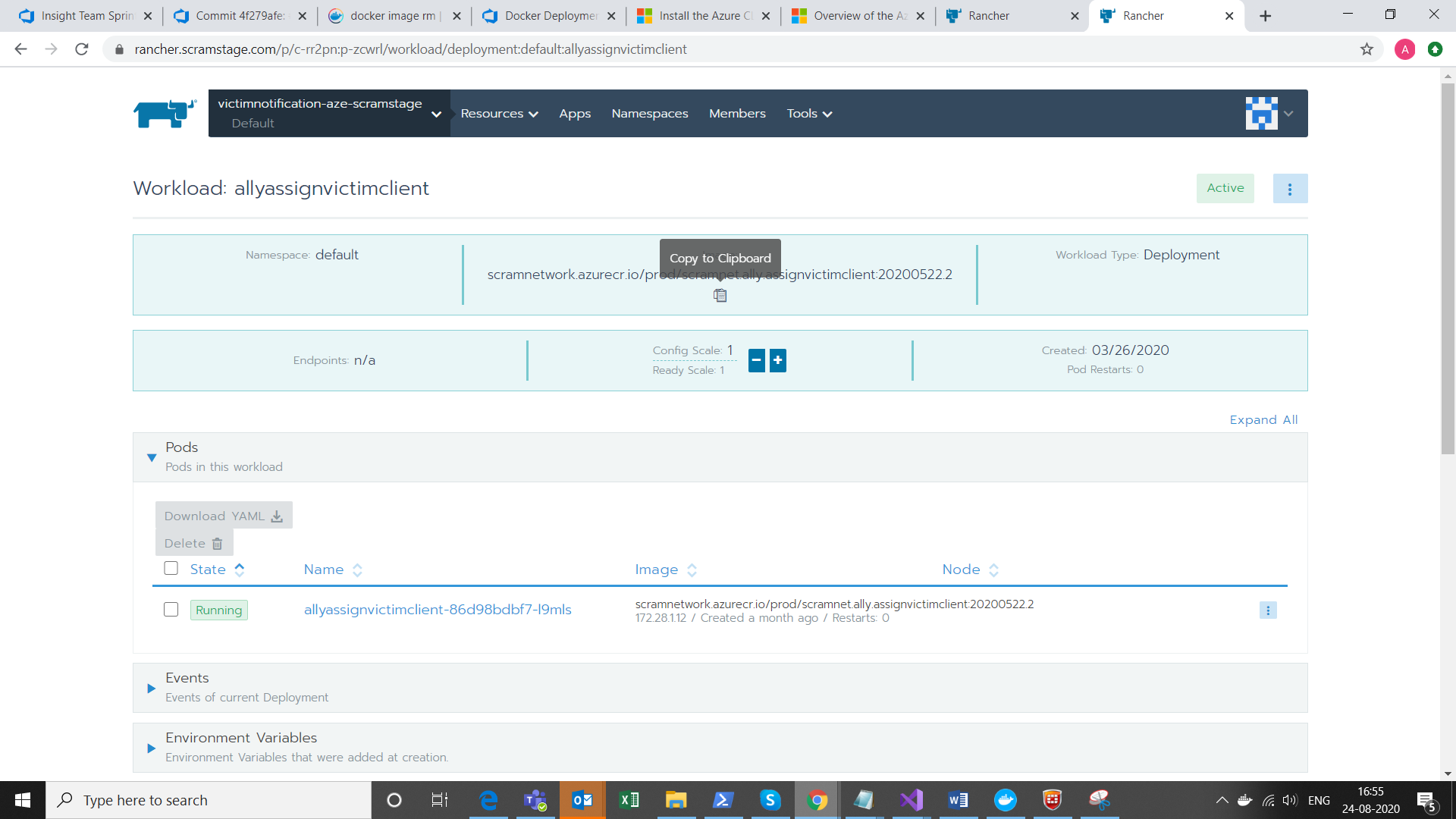


We see the list of all workloads

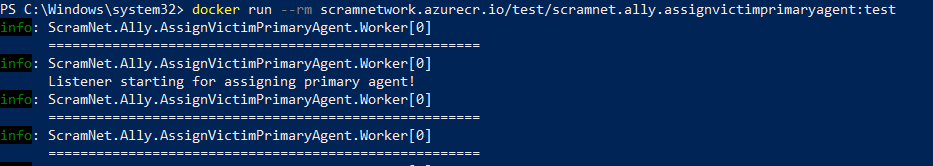


Click on any workload you wish to pull image for.

Copy the imagename



## Run the pulled image

Command : docker run --rm scramnetwork.azurecr.io/test/scramnet.ally.assignvictimprimaryagent:test

Reference link for this topic :

[https://alcoholmonitoring.visualstudio.com/DevSecOps/\_wiki/wikis/Infrastructure.wiki/231/Docker-Deployments?anchor=\*\*pull-and-run-images\*\*](https://alcoholmonitoring.visualstudio.com/DevSecOps/_wiki/wikis/Infrastructure.wiki/231/Docker-Deployments?anchor=**pull-and-run-images**)

# Reference links

* DevSecOps Wiki

<https://alcoholmonitoring.visualstudio.com/DevSecOps/_wiki/wikis/Infrastructure.wiki/231/Docker-Deployments>

* For troubleshooting issues :

<https://docs.docker.com/docker-for-windows/troubleshoot/>

* Docker official documentation

<https://docs.docker.com/>

