**Investigate whether it is possible to execute MSI files from a Docker container**

Proved the concept using UDP server-client and docker volume approach using dotnet core technology.

**Steps**

* Create a windows docker image using commands

1. Open command prompt in administration mode
2. Create new c# console application using **dotnet new console -n msiexecutor.0.1** command
3. Create Dockerfile without extension and add configurarion

FROM mcr.microsoft.com/dotnet/core/sdk:2.2 AS build-env

WORKDIR /app

COPY . .

RUN dotnet restore

RUN dotnet publish -c Release -o out

FROM mcr.microsoft.com/dotnet/core/aspnet:2.2

WORKDIR /app

COPY --from=build-env /app/out .

ENTRYPOINT ["dotnet", "msiexecutor.0.1.dll"]

1. Run

docker build -t msiexecutor.0.1 .

1. Create docker volume using below commands

docker volume create --name data

1. Check list of volums

docker volume ls

1. Write logic to download \*.msi file from network path and send notification to client which is running on host machine



1. Create another console application which is going to receive the message from server, it will run in host machine.



1. Run the docker image and receiver both

docker run --dns=8.8.8.8 -v data:c:\data msiexecutor.0.1 "10.0.75.1" <https://csg6dfc23c08ba8x4c21xb3e.blob.core.windows.net/msisetup/PM650FirmwareUpdate.msi>

which taken UPD ip and msi url