**Software Container**

Resource Conflict, Different version required for different application

Solution[Symbol]Virtualization

Host + Hypervisor[Symbol] Virtualize the hardware ,Create duplicate operating System, Duplicate files

Container Engine[Symbol]Virtualize the operating System[Symbol]quick iterative deployment of application with high success rate[Symbol]can ship not only my application  but also the application’s dependencies.

Docker:

A platform used to build and deploy container.

Image: An image is what Docker uses to create a container

[Symbol]Image is what we can create using Docker build command

[Symbol]Image will container application files and libraries as well as dependencies(.Net).

[Symbol]Docker will create one or more container using image as a template

[Symbol]Docker Registry is like package where we find Image.

[Symbol]No longer deploy dll or exe file rather we deploy an Image

[Symbol]Need to install Docker for windows

Windows 10, Microsoft Hyper-V installed,

After installing:

Right Click in right below corner[Symbol]

I can run both Windows and Linux container

Hyper-V manager: can open in start menu

We can switch to use windows container(by right click on docker):

Docker[Symbol]pull Image[Symbol]create container[Symbol]write program[Symbol]create a new image[Symbol]pass this image to anyone who has Docker running on his computer[Symbol]he can run without installing my application[Symbol]net framework

Check Docker status through CMD:

Check version of client and server

Change the server from Linux to Windows container

Can pull image from Docker store

Pulling an image that support asp.net core for windows server

Now instantiate this image into a container but before that see process status (PS) which shows current running container.

It—interactive terminal run command instantiate the container with name ‘firstapp’ using image id (using 1st 2 letter now).

After few moments we will see a command prompt for the container (not the machine that I am logged in to but the container on my machine).

We can see directory here in this container

Now flip over other command prompt and see the container status

Now in container make a new folder and check for dot net

Create new console application (same like create new project in Visual studio)

And now see the firstapp directory

After build our program is ready to run.

Now we make this program ready for others to run so[Symbol]move to directory[Symbol]echo our folder and run the path of program having a new name as hello.cmd in c directory. Now we can run this program.

Exit command will exit the container but all the changes will be saved.

Now we see the container by command ps –a

Now we create a new image from this container which contains all the changes, files and application.

If we need a container into a specific stage, Docker commit the container so that we can pull the container into new image

Here commit the container a8 into new image named pluralsight with tag firstapp.

And when we see the available image using Docker command, new created image will be there.

Pluralsight image only contain difference so I uses less memory only because dependencies are there where we build the application.

Now Docker run the Image having ID starts with 23 and command hello.cmd

I now have a container image that I can place into a Docker registry that could be public registry that anyone can pull from or could be a private registry that I setup for myself or my company. From the registry I can pull this image onto any machine that can run window container. I don’t have to worry if the machine has .Net core installed or >net SDK installed. My application is in the image that is ready to run because all the dependencies are in place.

And now I can use that image to spin-up the containers on local machine, in cloud, in cluster, everywhere.

Now Docker with Visual studio:

Enable Docker support checkbox

Change the configuration and application will run inside the container.

Docker Compose is a tool which helps to manage different container for different application (suppose an application contains API, Web application and database then we should three containers for the three services (giving single responsibility to each container).

.yml file[Symbol] now has one service only and one container[Symbol]service name as pswebapi, image name and build context.

Now we want to spin up our container to the cloud so that anyone can access our API

In Azure[Symbol]

In VM[Symbol]Create container registry[Symbol] can push the image from Docker to here[Symbol]Image will be close to the server[Symbol]deployment will be bit faster

If we are going to deploy container in Azure Create container registry will be worth.

Can easily push or pull Image from the registry.

Now backing out of this and coming back to the container search[Symbol]Azure container service[Symbol]for complex system with multiple container and easily scale up and down the container[Symbol]it can manage cluster of virtual machine

[Symbol]we can use service fabric to deploy the container

Now container and APP service:

Create web app with container[Symbol]either add image in private registry in Azure or in Docker Hub[Symbol]

Create Public Repository in Docker Hub[Symbol]

[Symbol]In Visual studio change Image name to the following

[Symbol]

Now will use Docker push and Docker will understand that where this should live, sallen/pswebapi.

And when build this image we want to send a release version or release build version on cloud to run .

 Now we can see sallen/pswebapi Image

Visual studio has tool where we can just check in code in VSTS and VSTS able to push the build in Docker Hub or in Azure Container Registry for continuous build.

Here we use manually now[Symbol]login and push command[Symbol]

Can see the latest build here[Symbol]

Configure Docker Hub here[Symbol]

[Symbol]create and done , we have running app service

 Now for continuous build[Symbol]

1. Set value as key value in application settings

Key=

Value=true.

      2. Create a webhook in DockerHub site as

Give web hook name and webhook url and url will point into Project Kuddu Site

Webhook URL: (username password from GetPublishProfile in Azure overview blade)

 /docker/hook

Now make some change in code and build[Symbol]and push the build in Docker Hub[Symbol]webhook see the latest build and push the image in Azure and publish the web API.

Container[Symbol]we deploy both Application and Application deployment Environment

Containers  are lightweight compared to virtual machine . we can get more from exicting hardware while our application are still isolated from each other.