Azure Quick Walk Through’s.

Selected Azure Services.

1. Azure container-related Services.
2. Azure Resource Management Template (ARM Template)
3. Cicd
4. Dns
5. …..
6. …….

Azure Container Services:

Case Scenario:

Assume your organization or you want to use Azure to run its web apps. For this reason, it makes sense to store the images in **Azure Container Registry**, then run them using the **Azure Container Instance service.**

Azure provides a wide range of services that help you to work with containers. These key services include:

- Azure Container Instance (ACI)
- Azure Container Registry (ACR)
- Azure App Service
- Azure Service Fabric
- Azure Kubernetes Service (AKS)

For now, we will focus on **ACR** and **ACI.**

Azure Container Registry (ACR).

ACR is a private registry for hosting container images. Using the Azure Container Registry, you can store Docker formatted image for all types of container deployments. Azure Container Registry integrates well with orchestrators hosted in Azure Container Services including Docker Swarm, DC/OS, and Kubernetes.

Use Azure Container Registry to:

- Store and manage container images across all types of Azure deployments
- Use familiar, open-source Docker command line interface tool
- Keep container images near deployments to reduce latency and costs
- Simplify registry access management with Azure Active Directory
- Maintain Windows and Linux container images in a single Docker registry.
-

Azure Container Instance (ACI).

ACI offers the fastest and simplest way to run a container in Azure without having to manage any virtual machines or adopting a higher-level service. ACI enables deployment of Docker containers onto Azure Infrastructure.

ACI is a recommended solution for scenarios that can operate in isolated containers which includes simple applications, task automations and build jobs.

For the docker to be deployed using ACI, the image needs to pull from a private container registry, such as ACR or Docker Hub.

Some features of ACR:

- Fast Startup times
- Public IP connectivity and DNS name
- Hypervisor-level security
- Custom Sizes
- Persistent Storage.
- Support Windows and Linux containers

Let see how this works in the very basic concept.

This Walk through would show how to publish a Docker image to **Azure Container Registry**, and run an image using the **Azure Container Instance service.**

Two main tasks to complete:

1: Publish/store local docker Image to an Azure container Registry (ACR)

2: Run the new docker Image from ACR to Azure Container Instance (ACI)

1st task:

Publish (local) Docker Image to Azure container Registry (ACR)

Things you need:

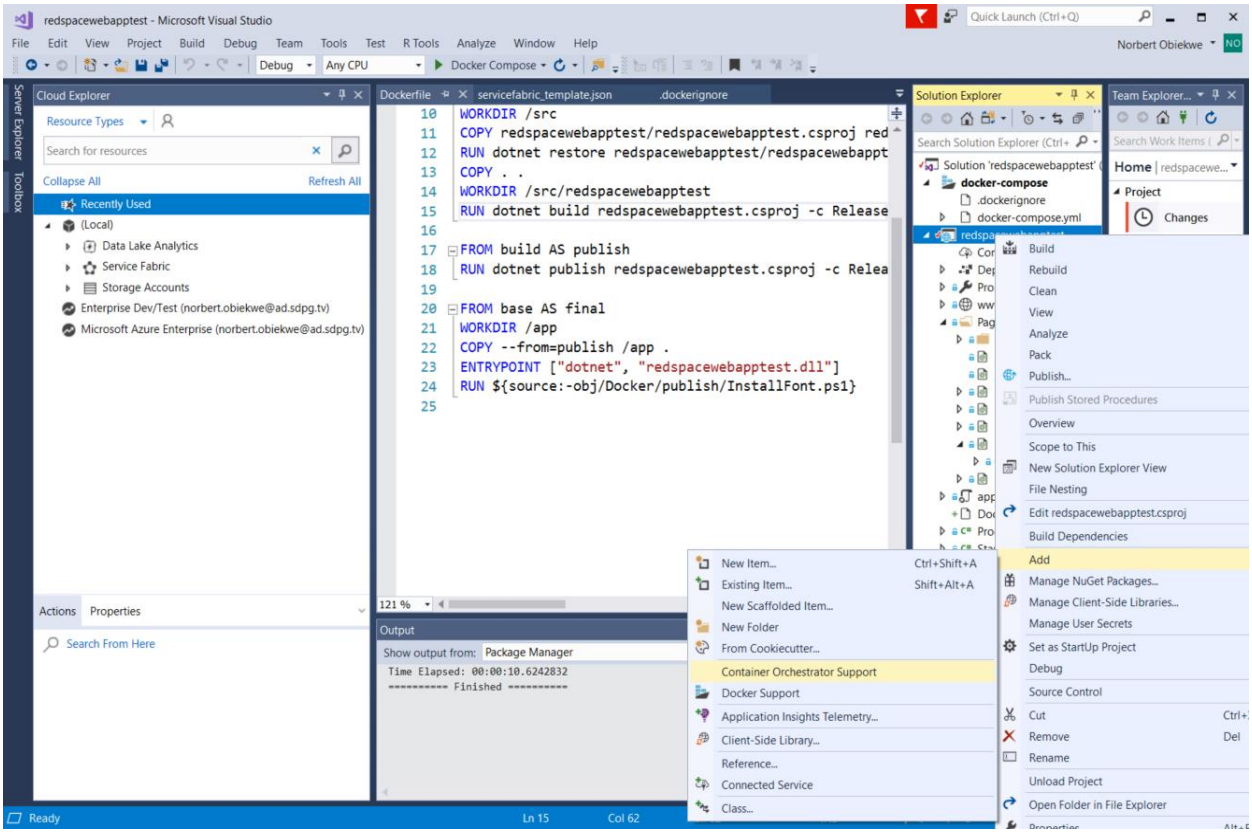
- Active Azure subscription
- Local running docker App
- Visual studio, Code or Azure Command line Interface (CLI)

Publishing a local docker to ACR can be done either through a **command line interface (CLI) or using the publishing Wizard** in Visual Studio or Code or any other GUI with an Azure extension support.

I prefer the CLI method (using Linux OS based image).

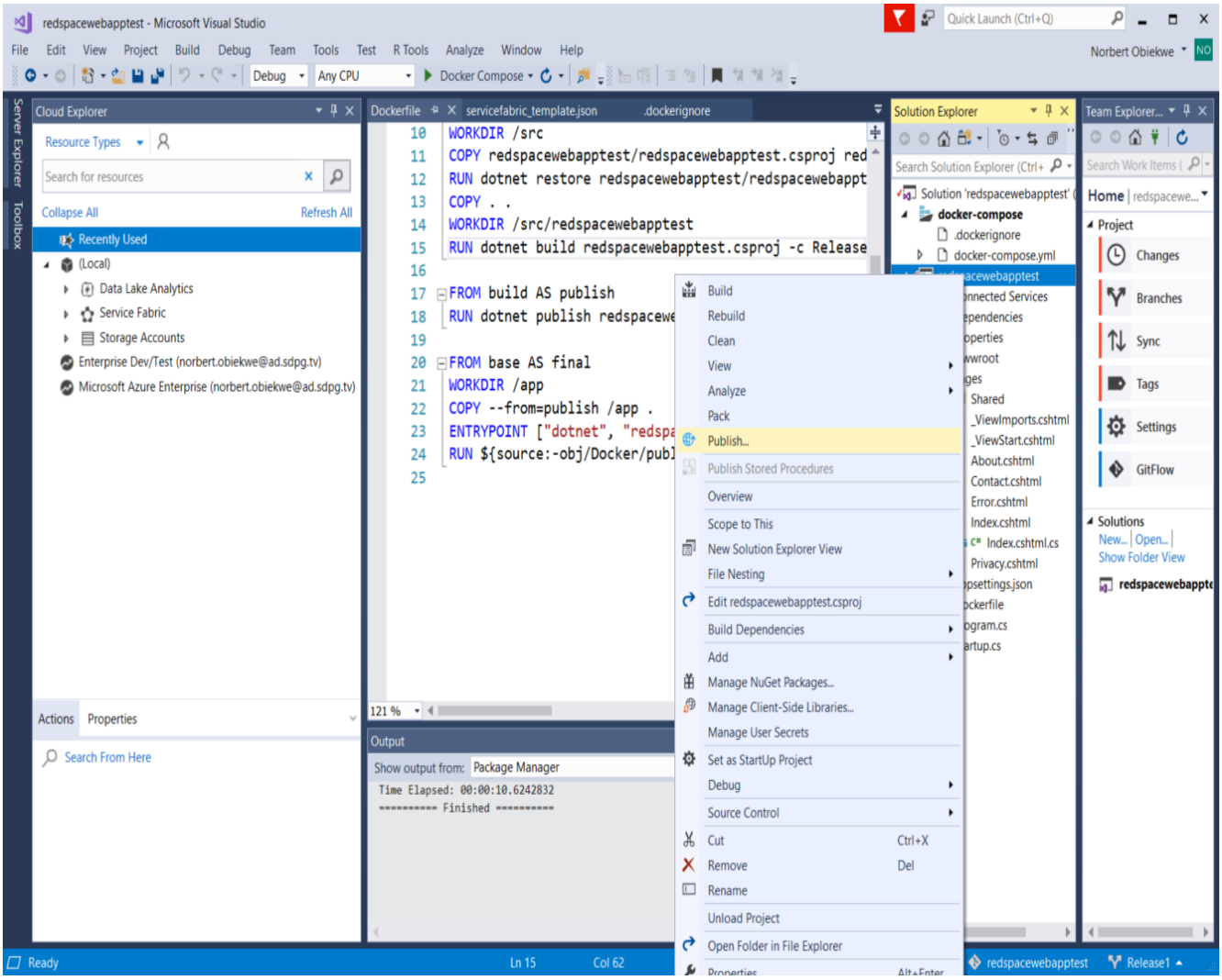
In this walk through, you would be publishing your application that is running successfully in local Docker to **Azure Container Registry** using the **publish Wizard** method. **(It is assumed you have a running Dockerized App on your local machine).**

This walkthrough would use a sample dockized App (redspacewebapptest) in Visual Studio and push it to ACR. *If the app is not containerized yet using Docker, right click on the project, select **ADD > Container Orchestrator Support (target container for this guide is OS Windows). Linux is highly recommended for CLI**

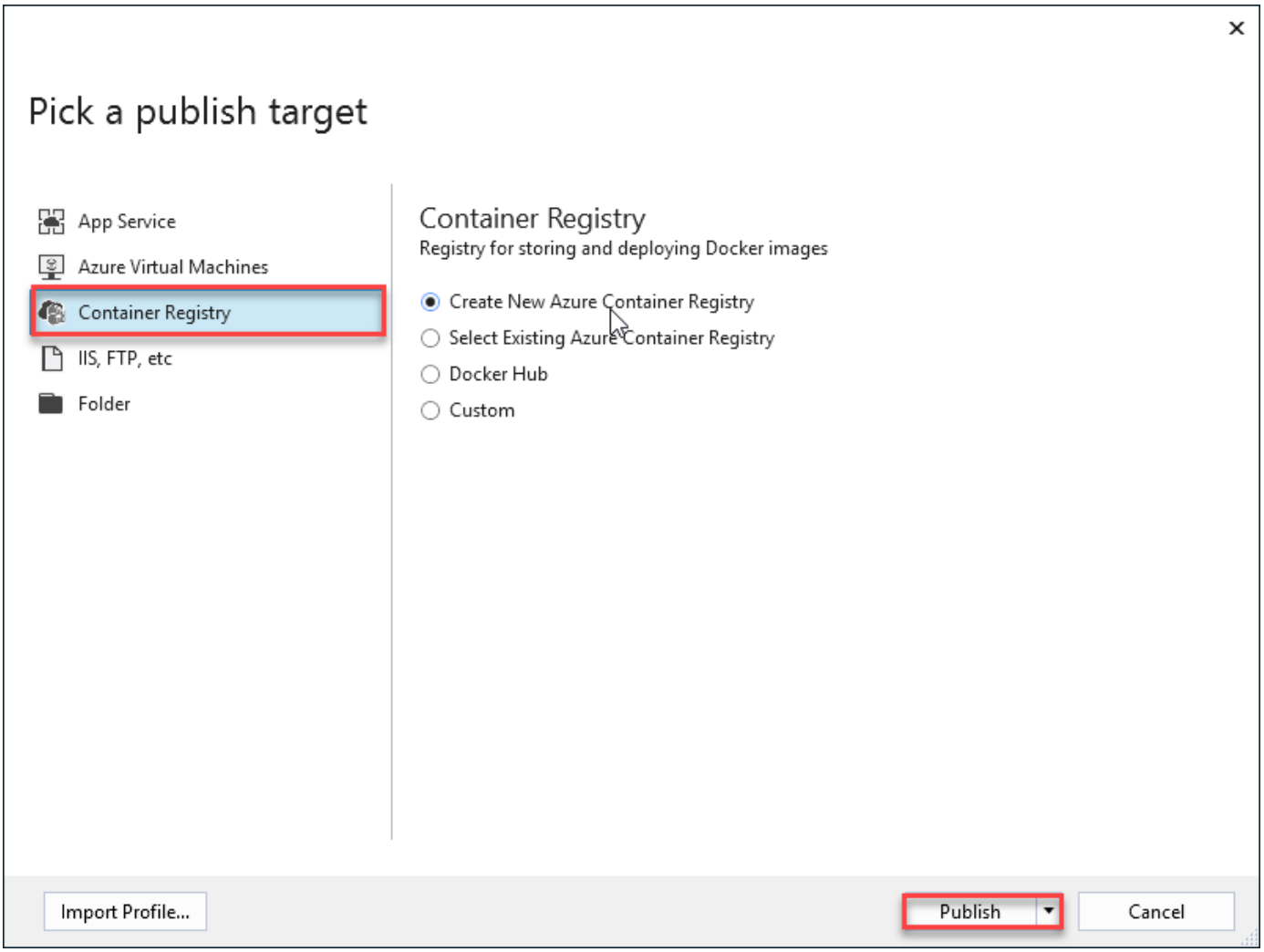


Steps to publish local docker to ACR: (6 steps, GUI method)

1. Right-click on the project, select Publish

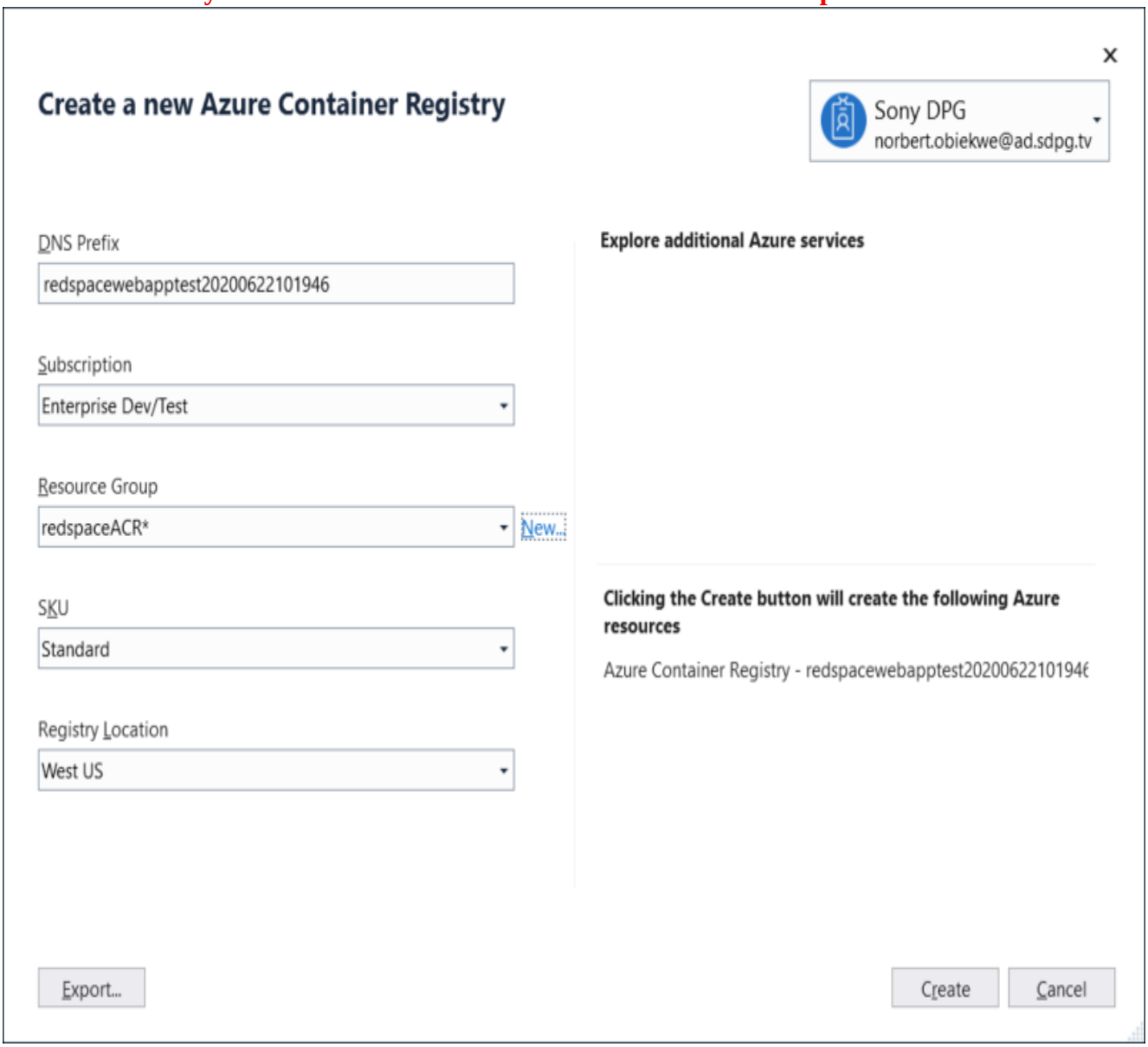


2. In the publish wizard select **Container Registry** and select **Create New Azure Container Registry** and click on **Publish**

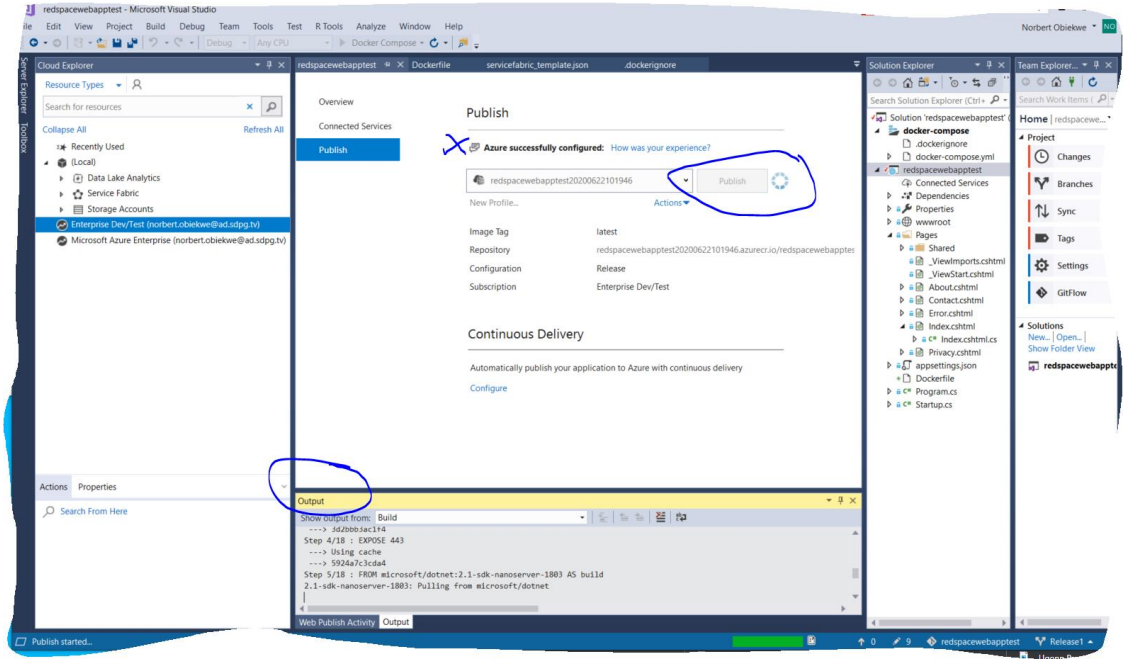


3. Fill the required details and click on **Create** (Alternatively, you could first create your ACR in your Azure subscription, then Visual Studio would pick up your existing ACR's for you to select from) **See the create container registry in Azure video below.**

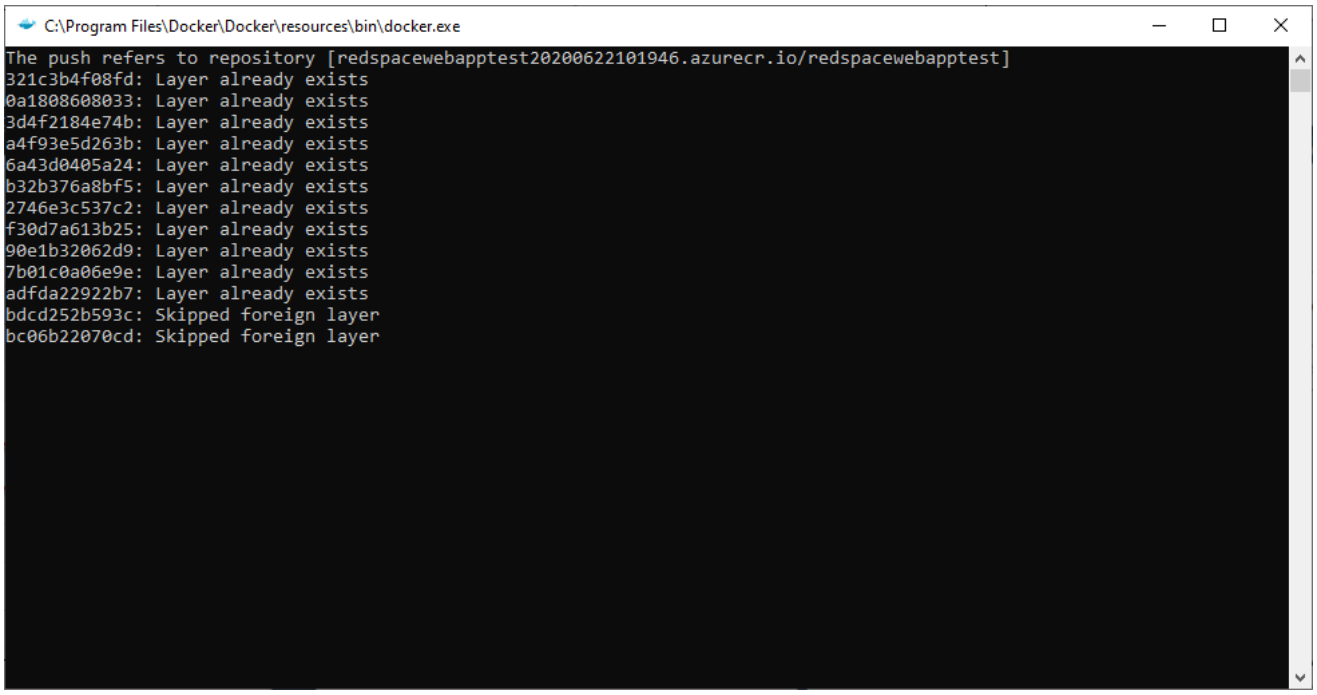
before creating any resource in Azure, you would first create a Resource group. Every resource must be associated with a Resource Group

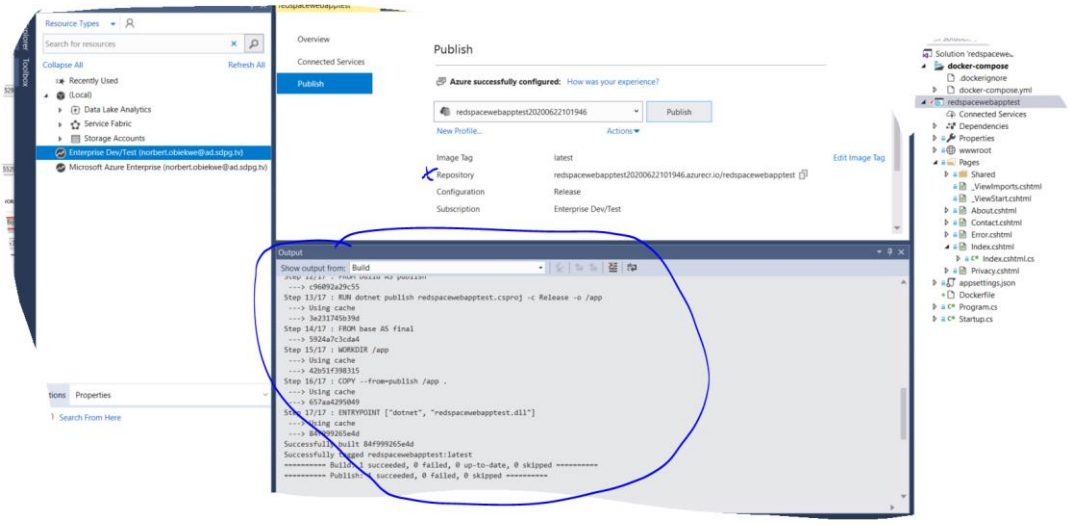


On create, you should observe the publish running as shown.



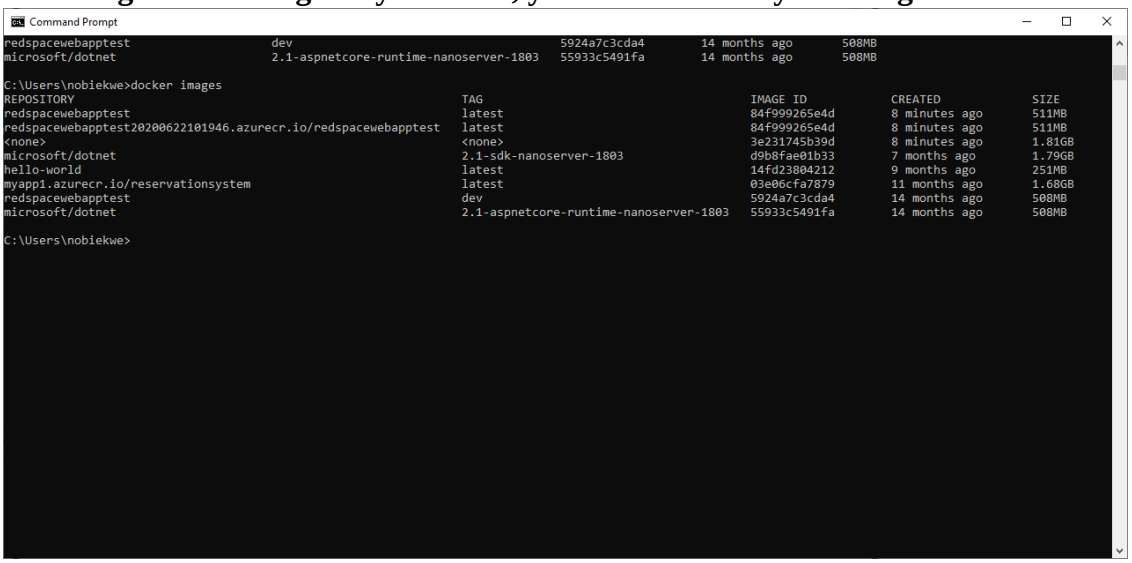
4. **When publishing, the production Docker image is created and pushed to the Azure Container Registry.**





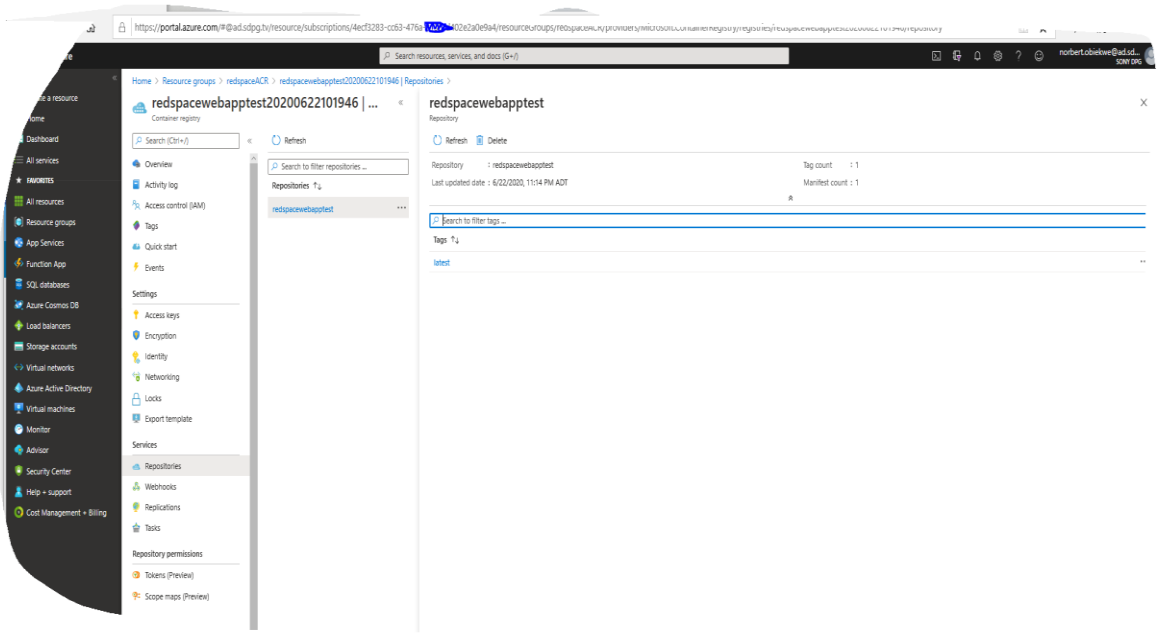
Observe the Output console for error or if ACR deploy is successful

Checking docker image on your local, you can see list of your images

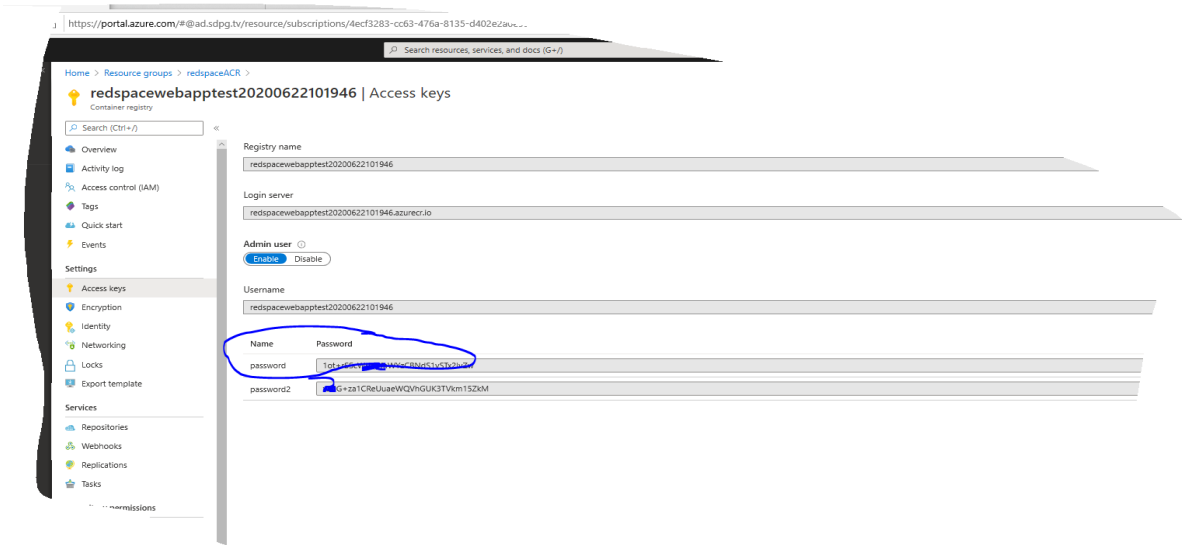


```
Command Prompt
redspacewebapptest                    dev                                 5924a7c3cda4        14 months ago       508MB
microsoft/dotnet                      2.1-aspnetcore-runtime-nanoserver-1803   55933c5491fa        14 months ago       508MB

C:\Users\nobiekwe>docker images
REPOSITORY                                                          TAG                                 IMAGE ID            CREATED             SIZE
redspacewebapptest                                                  latest                              84f999265e4d        8 minutes ago       511MB
redspacewebapptest20200622101946.azurecr.io/redspacewebapptest      latest                              84f999265e4d        8 minutes ago       511MB
<none>                                                              <none>                              3e231745b39d        8 minutes ago       1.81GB
microsoft/dotnet                                                    2.1-sdk-nanoserver-1803             d9b8fae01b33        7 months ago        1.79GB
hello-world                                                         latest                              14fd23804212        9 months ago        251MB
myapp1.azurecr.io/reservationsystem                                 latest                              03e06cfa7879        11 months ago       1.68GB
redspacewebapptest                                                  dev                                 5924a7c3cda4        14 months ago       508MB
microsoft/dotnet                                                    2.1-aspnetcore-runtime-nanoserver-1803   55933c5491fa        14 months ago       508MB

C:\Users\nobiekwe>
```

5. Once the publish is successful, navigate to the deployed ACR in Azure portal to confirm your published image to ACR. On your container registry Account, go to repositories to see your repository and image tag



6. **Note the image registry password, you would use the password for next exercise (pushing the docker image from ACR to ACI)**



At this point, you have successfully published/stored your local docker image to your Azure container Registry. To put the image into use (accessible publicly) you must push the image from ACR to ACI.

Publishing your local docker to ACR can as well be done using Azure Command line interface (Cli), instead of GUI.

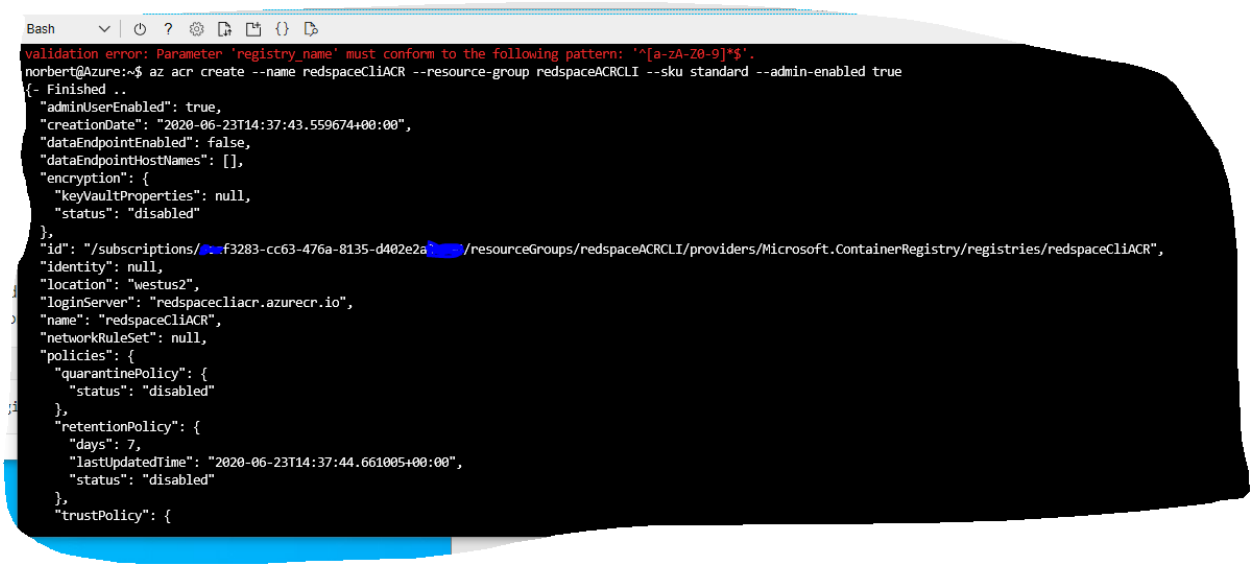
Here is walk through to publish an image to ACR using the cli.

What you need:

- **Azure Account**
- **Azure CLI bash or local azure cli**

Steps:

1. **Create Azure container registry in Azure using the az acr create command or through the azure portal.**



az acr create --name redspaceCLiACR –resource-group redspaceACRCLI --sku standard –admin-enabled true

Or create an ACR in Azure portal

- Sign in to the Azure portal with your Azure subscription.
- On the Azure portal menu or from the **Home** page, select **Create a resource**.
- Select **Containers**, and then click **Container Registry**.

you need to create a new resource group or use an existing resource group

- Different SKUs provide varying levels of scalability and storage.
- Azure Container Registry repositories are private — they do not support unauthenticated access. To pull images from an Azure Container Registry repository, use the docker login command and specify the URL of the login server for the registry.
- The login server URL for a registry in Azure Container Registry has the form <*registry_name*>.azurecr.io.

1.1 Authenticate to docker using docker login <yourregistryname>.azurecr.io

```
norbert@Azure:~$ docker login redspaceCLiACR.azurecr.io
Username: redspaceCliACR
Password:
WARNING! Your password will be stored unencrypted in /home/norbert/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
norbert@Azure
```

. To get your docker login credentials run.

-- Az acr credentials show –name <yourregistryname>

Upon creation of ACR, you then publish your local docker to ACR

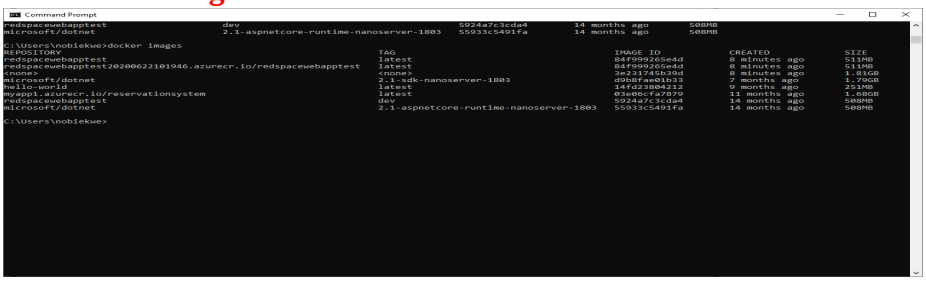
2. **Upload the local docker image to ACR: the image needs to be tagged before pushing it to Acr**

In your local command line, run the following command to tag your local image with the name of your registry. Replace <registry-name> with the name of your registry in Azure Container Registry.

- **docker tag <yourimage>: latest redspaceCLiACR azurecr.io/<yourimage>:latest**

run the docker image ls command to verify that image has been correctly tagged

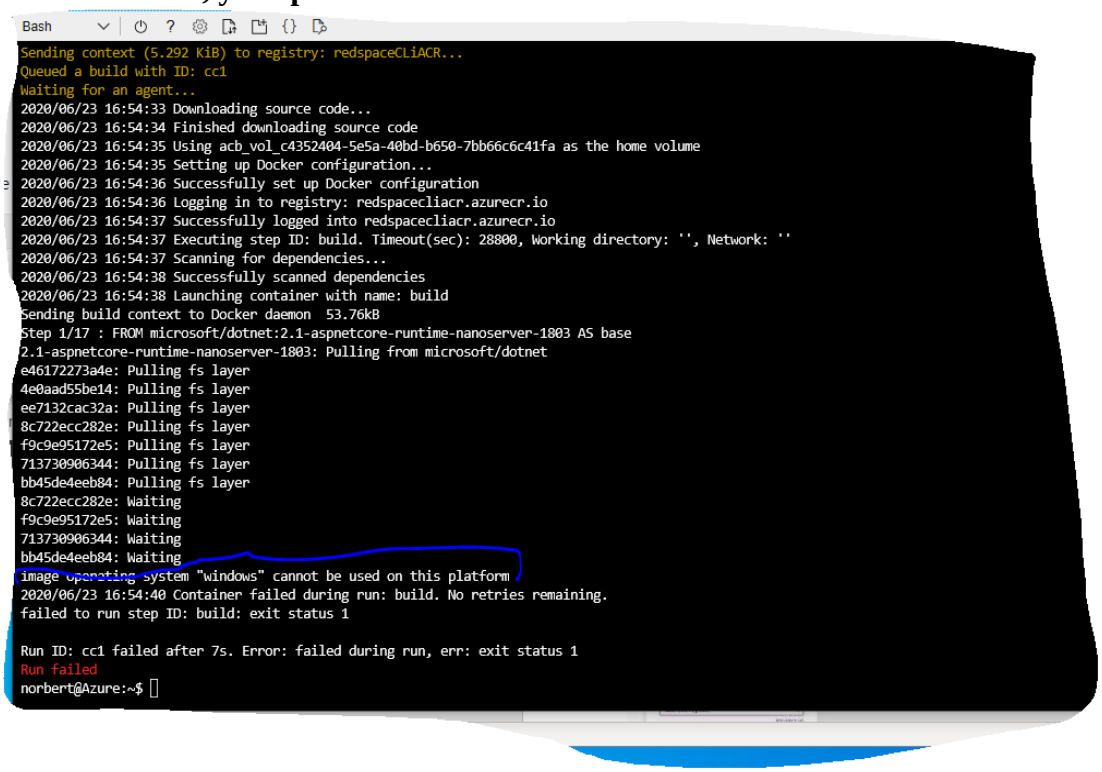
- **Docker image ls**



- **Push your image to ACR –**
  (You must login and authenticate first by using the docker login <login-server>)
- **On your bash, run the docker push command**

- **Docker push redspaceCliACR.azurecr.io/redspacewebapptest : latest**

If using CLI to push to ACR, note that your image operating OS should be Linux. If your OS is windows, your push to Acr would fail shown: You could use PowerShell instead



Alternatively (if your image OS is windows like here) use the publishing wizard method as discussed above

2nd task.

Run the published docker image in ACR using ACI

In the previous steps, you published your docker image to ACR. You now want to make the app available globally. You ‘ll use the azure container instance to run the image.

Azure Container Instance enables you to run a Docker image in Azure.

What you need:

- **Azure subscription**
- **CLI Batch**

1. Execute the following **az container** create command to deploy a container instance and push the redspacewebapptest (your published app) image from ACR.
   Replace <username>,<password> in the following command with your registry's admin username and password.
   Replace <location> with the location value returned when you created the container registry earlier., (you can also get the location from your acr overview on Azure portal)

   **az container create **
   **--resource-group <your resource group name> **
   **--name acr-tasks **
   **--image <myregisstryname>.azurecr.io/<my image>:v1 **
   **--registry-login-server <myregisstryname>..azurecr.io **
   **--ip-address Public **
   **--location <location> **
   **--registry-username [username] **
   --registry-password [password]

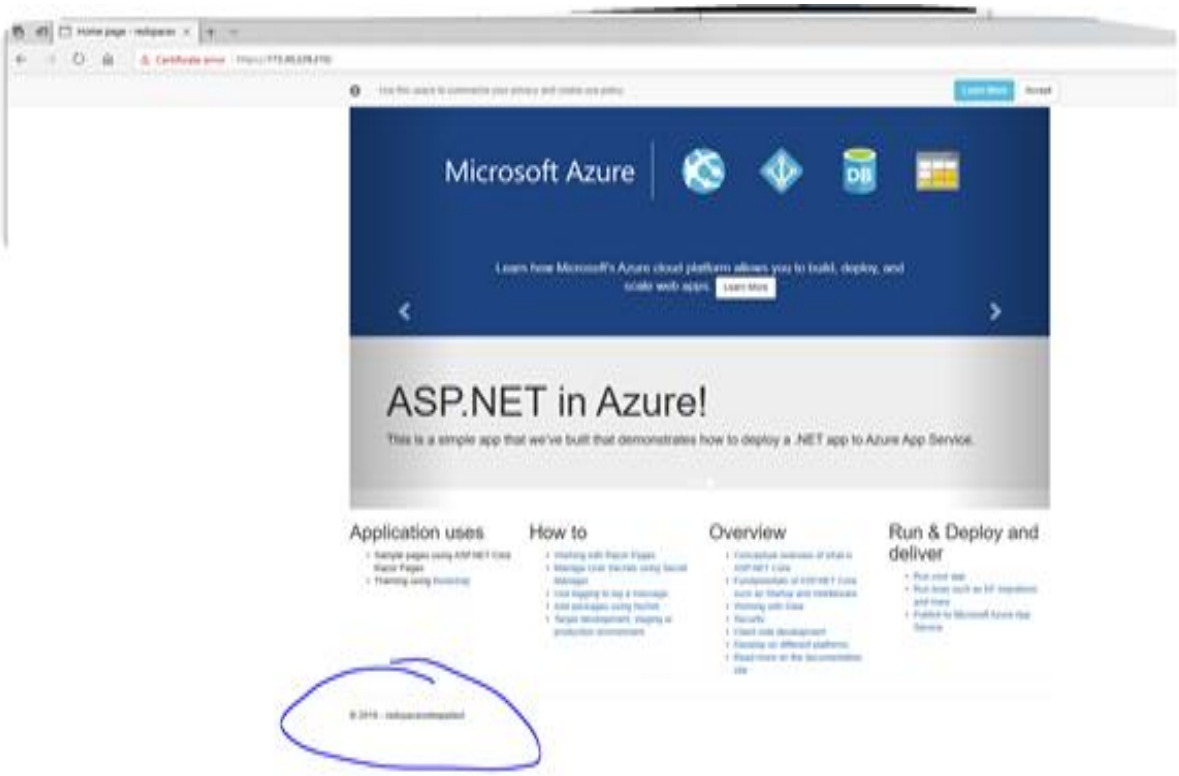
```
norbert@Azure:~$ az container create --name redspacecontainer --resource-group redspaceACR \
> --os-type windows \
> --image redspacewebapptest20200622101946.azurecr.io/redspacewebapptest:latest \
> --ip-address public
Image registry username: redspacewebapptest20200622101946
Image registry password:
```

- **Use your image registry password, accessible from your Access key on ACR**
- **Enter your acr name and image**
- **It might take about 5 -10 approximate minutes to deploy**
- **Navigate to your resource group where the ACI is deployed, select the image and locate the IP address.**

To access your App, Get the IP address of the Azure container instance using the following command.

- az container show --resource-group <my resource group> --name acr-tasks --query ipAddress.ip --output table

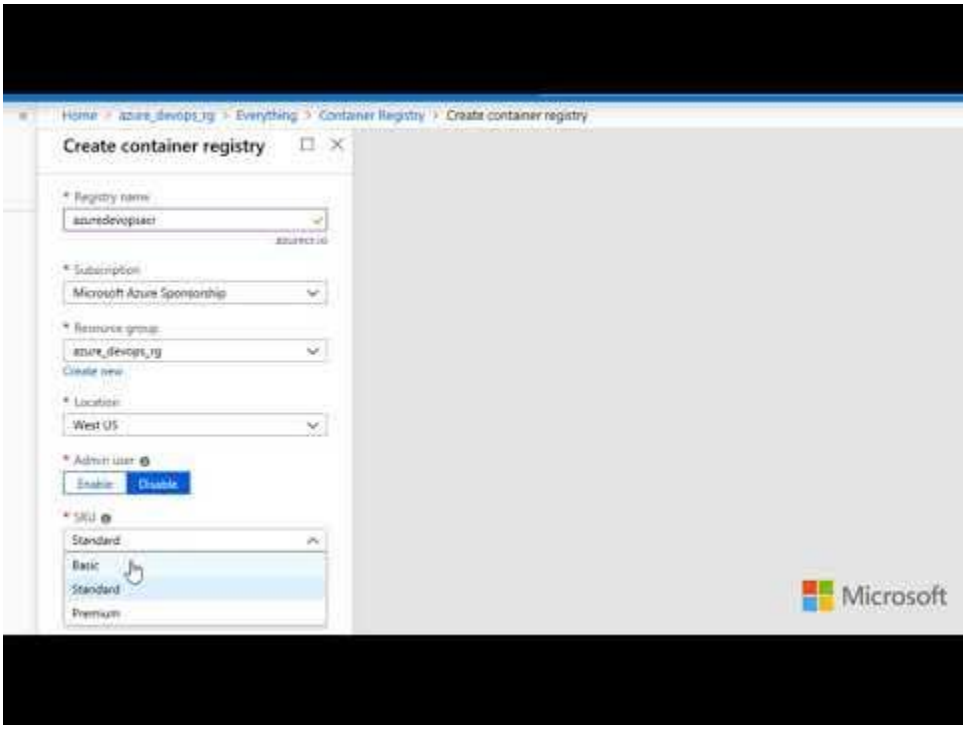
2. Open a browser and navigate to the IP address of the container. If everything has been configured correctly, you should see your app



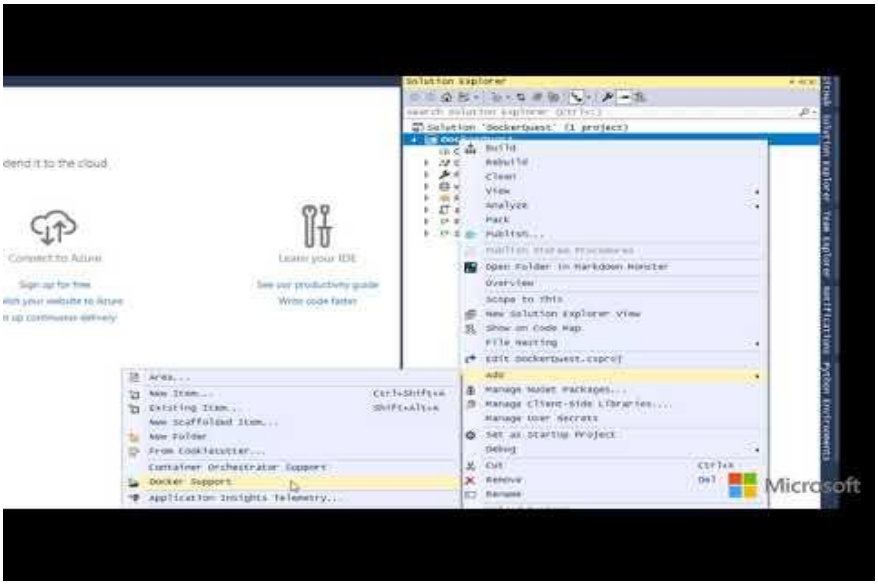
***If your image OS is Windows, it pays to have supported OS versions: Supported versions are Windows Server 2016 - Before 2B, Windows Server 2019 - Before 2B, Windows Server 2016 - After 2B, Windows Server 2019 - After 2B**

Summary: In this Azure Container Services walkthrough 101, you uploaded a Docker image to **Azure Container Registry**, and run same image using the **Azure Container Instance.**

Some might prefer a video guide: here is a walk-through video for ACR and ACI



Create Container Registry in Azure



Push image to the created Azure Container registry