**ACR**

Dev team packages the Application, lib, bin, other config files into a container image. This image will be pushed into container registry either private ( such as Azure container Instances) or public registry ( such as docker hub ). Using the azure portal, we deploy this image from the repo to azure container instances. ( which is pass ( platform as a service )). Admin work like upgrading, patching, infrastructure will be done by azure. We just need to maintain code and containers.

ACR allow us to upload the container and then the service will run the container, without worrying about underline infrastructure.

--------------------------------------------------------------------------------------------------------------------------------------

**Docker File:**  
  
FROM node:18-alpine AS installer

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

RUN npm run build

FROM nginx:latest AS deployer

COPY --from=installer /app/build /usr/share/nginx/html

Create ACR in azure portal. Create a container Registry in azure portal.   
  
Do this with basic plan, it will allow you to create in public.   
  
Once deployment is done, go to access keys and enable the admin user. This will generate the admin user name and password for admin, which will be used in azure devops pipeline.   
  
  
Now create pipeline, select **Docker ( pull and push image option )** and it will ask subscription as to create a service connection.   
  
  
default pipeline will be created. But we need some extra steps to be added in this.

Now we have to create azure container instance and deploy the image into that to run as container.  
  
Create a new task again with “ **azure cli** “ to actually provision the container instance and deploy the container on that.

-g is resource group name

--image is container name  
--dns-name-label  >> this is dns label on which your app will be available on the public ip. Give it a unique name

- task: AzureCLI@2

      inputs:

        azureSubscription: **'Pay-As-You-Go(0a386d52-f2c2-4fd8-97df-413412cc5762**)'

        scriptType: 'bash'

        scriptLocation: 'inlineScript'

        inlineScript: |

          az container create \

          --name **test**\

          -g **DevOpsTest** \

          --image $(containerRegistry)/$(imageRepository):$(tag) \

          --registry-login-server $(containerRegistry) \

          --registry-username **DockerTest11** \

          --registry-password **7zWwKjeAwPfc4ir8YK8Vqdds/y4B3lNJ9k8rCXYVAH+ACRA84Qtu** \

          --dns-name-label **demo-dockertest**

Open the azure container instances ( with the name test which is provided into –name )