Software Deployment

Creating containerized DevOps Pipeline that provides Node.js application in AKS

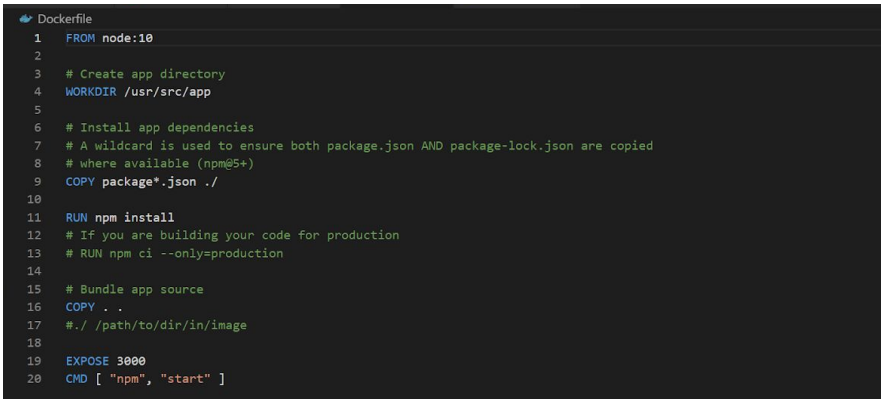
Links:

1. Link to Github Repo: https://github.com/KatjaFox/SoftwareDeploymentLab2
2. Link to the Docker Image in the Registry: nodejsnewcontainerregistry.azurecr.io/katjafoxsoftwaredeploymentlab:50
3. External IP address to see the deployed app: http://52.142.33.152:3000/

Steps:

Based on tutorial:
https://docs.microsoft.com/en-us/azure/devops/pipelines/ecosystems/kubernetes/aks-template?view=azure-devops

1. Create dockerfile in your existing node.js app so that a docker image could be build
   (Base for the docker file can be taken g.e. from https://nodejs.org/de/docs/guides/nodejs-docker-webapp/)
   Port was changed to 3000, as there is the node js app running internally.
   The CMD command was changed to “npm start”



2. Create a new azure container registry for your future project.

   In Azure CLI:

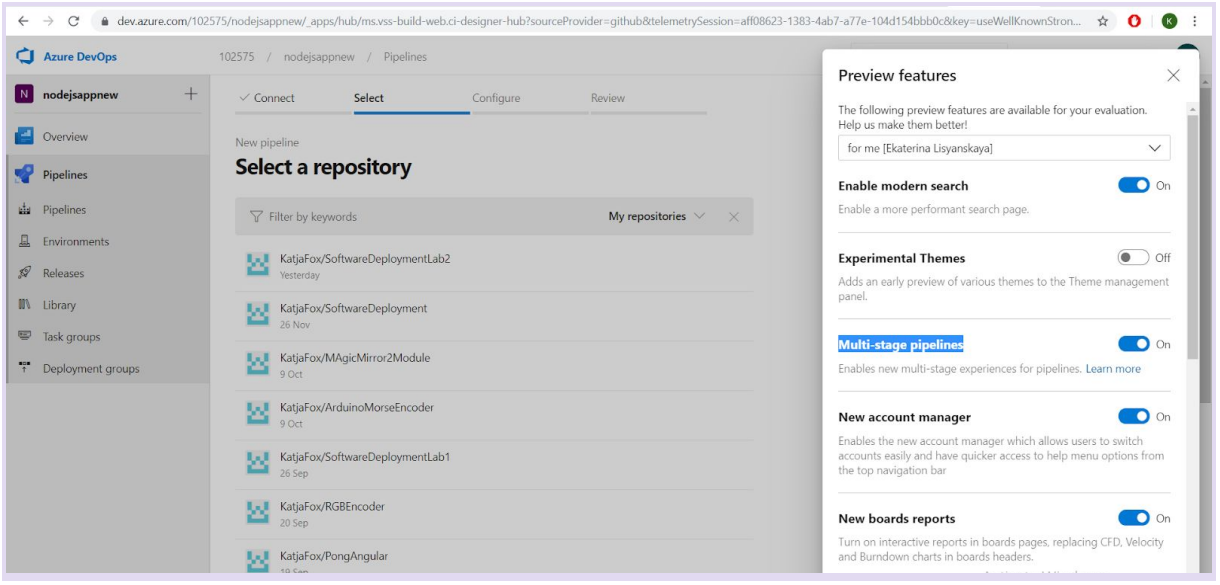
```
# Create a resource group
az group create --name nodejsnewapp-rg --location eastus

# Create a container registry
```

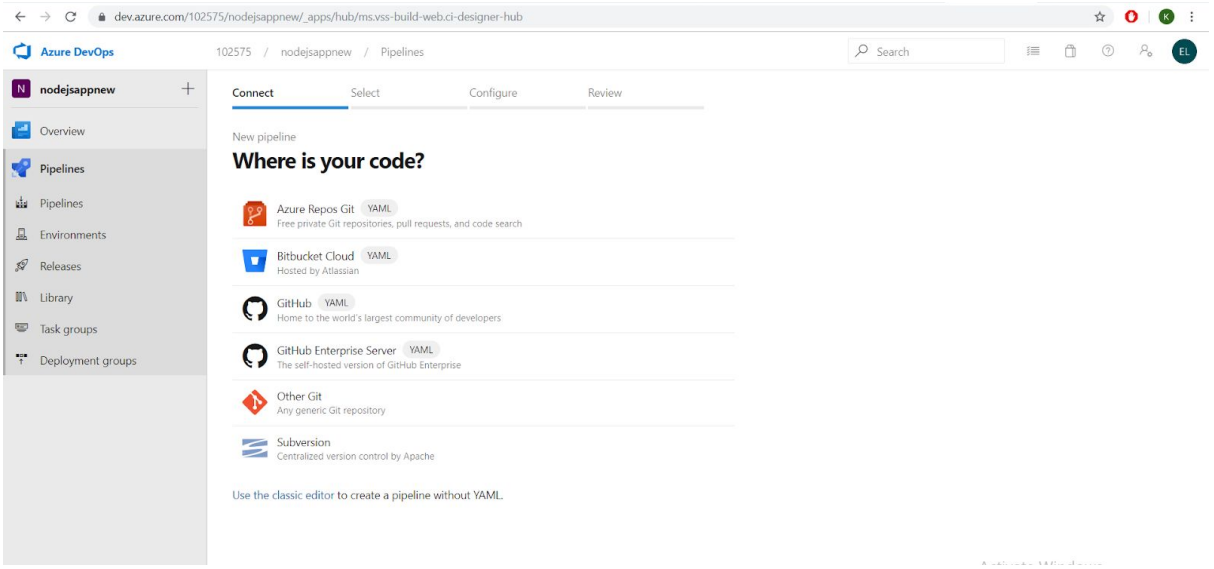
```
az acr create --resource-group nodejsnewapp-rg  --name
nodejsnewContainerRegistry --sku Basic

# Create a Kubernetes cluster
az aks create --resource-group nodejsnewapp-rg --name nodejsnewapp
--node-count 1 --enable-addons monitoring --generate-ssh-keys
```

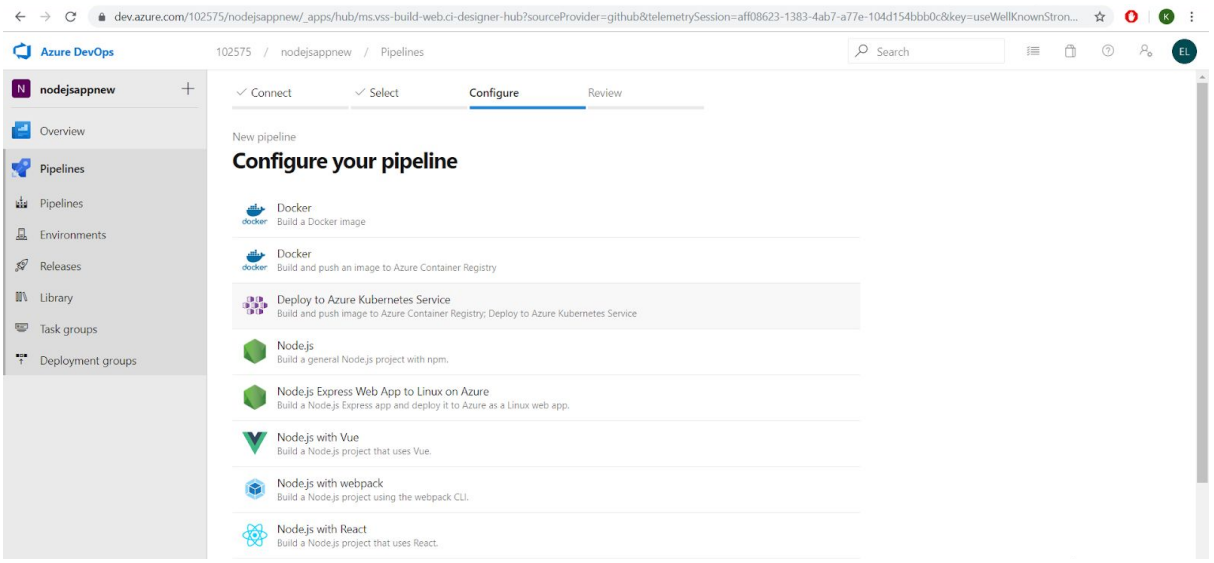
3. Turn on the Multi-stage pipelines experience for using a special template. In Preview Features of your account.



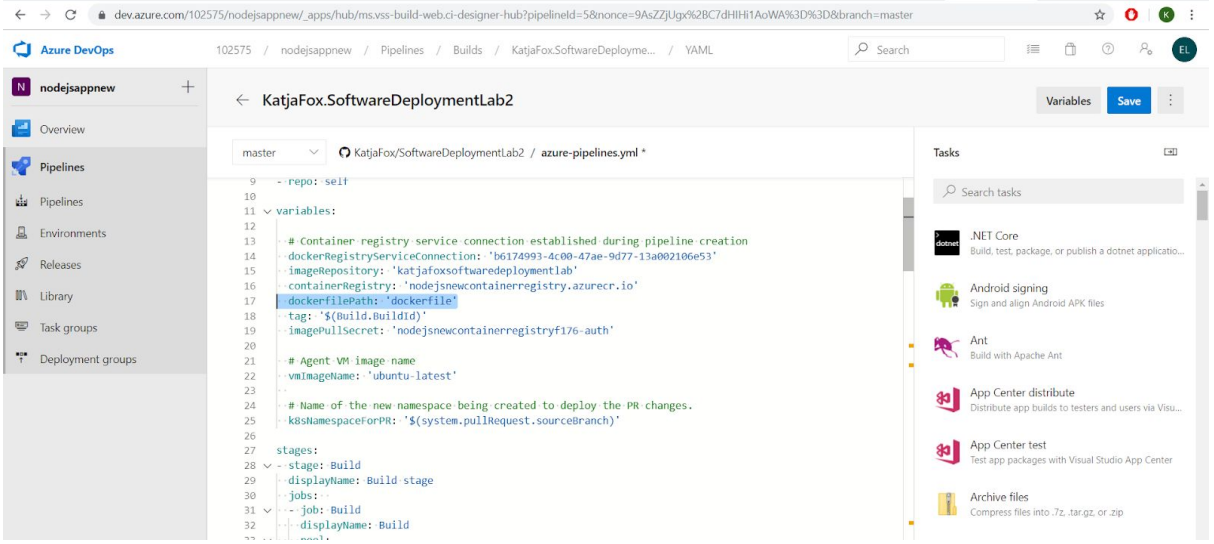
4. Within your selected organization, create a new DevOps project
5. In Pipelines select New Pipeline and GitHub as the location of your source code



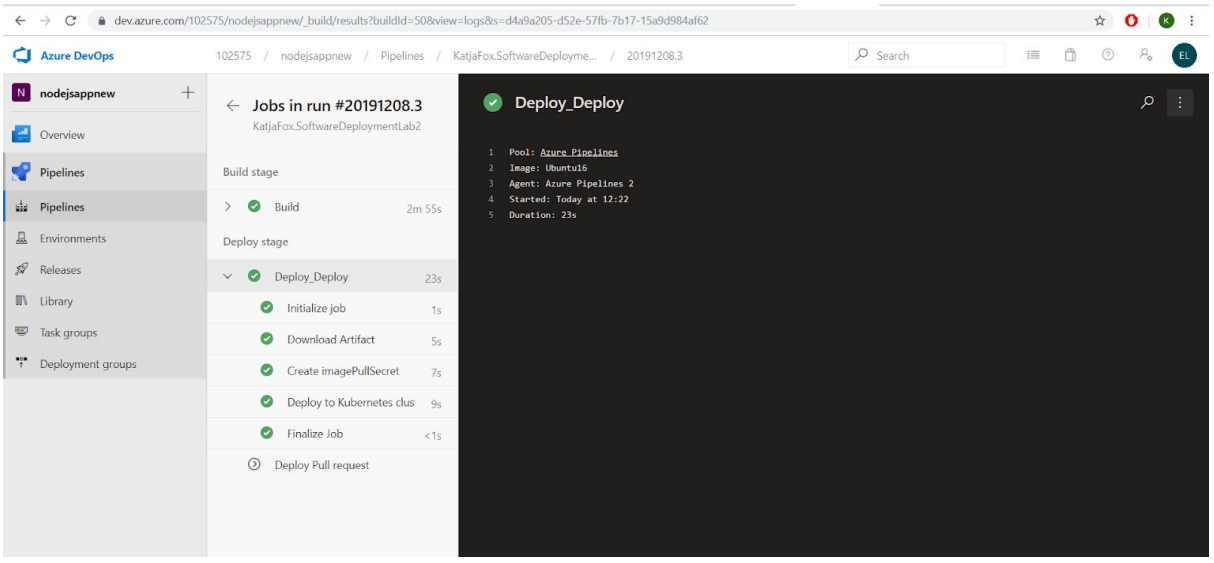
6. In Configure tab select Deploy to Azure Kubernetes Service.



7. For Namespace select Existing.
8. Set the “Enable Review App for Pull Requests” checkbox for reviewing app configurations in generated YAML.
9. Review the YAML to see what it does. Check that location of dockerfile is correct (not **/Dockerfile but your path)

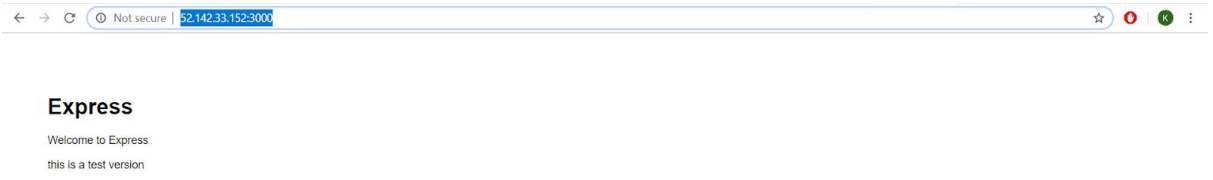


Select Save and run.



10. After deploying go to Envitonment/Services to see the external IP address where the app is deployed.

    To see the app running open: <IP address>:3000



What happened:
Creating the pipeline Azure Pipelines:

- Creates a Docker image and Docker registry service connection so that the pipeline can push the image into your container registry.
- Creates an environment and a Kubernetes resource in the environment.
- Generates an azure-pipelines.yml file which defines your pipeline.
- Generates Kubernetes manifest files. These files are generated by combining the deployment.yml and service.yml templates based on selections you made.
- Deploys the image in the AKS
- Publishes the app under external IP address