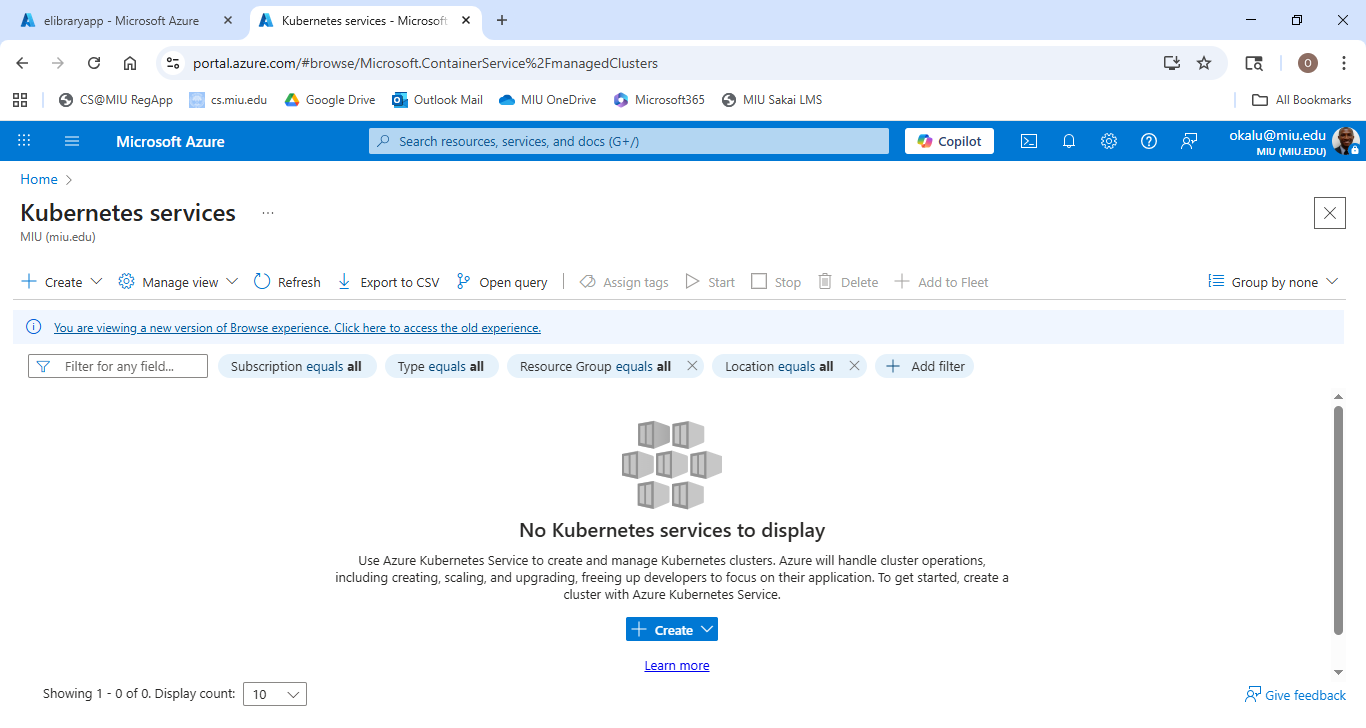
Demo: Deploy Spring WebApp to AKS

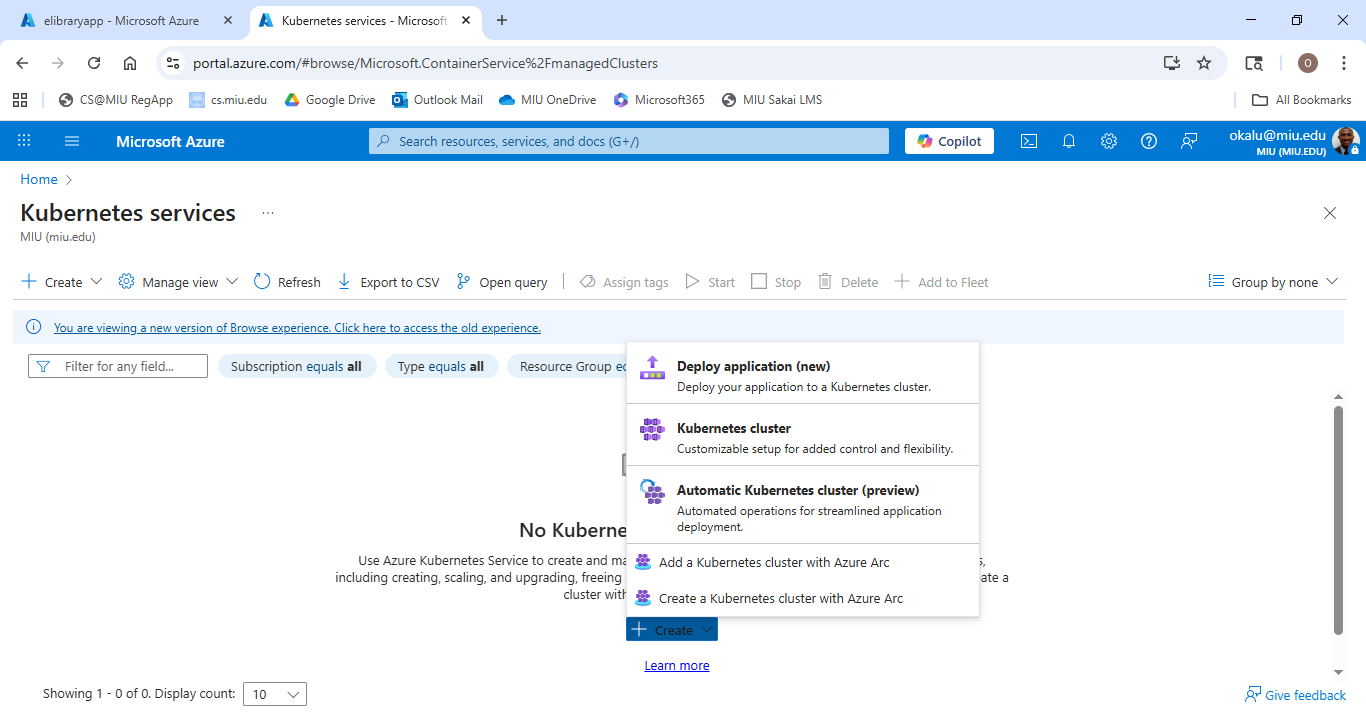
1. Obtain and install Azure CLI
   1. Using Window Package Manager (Winget), run

> winget install --exact --id Microsoft.AzureCLI

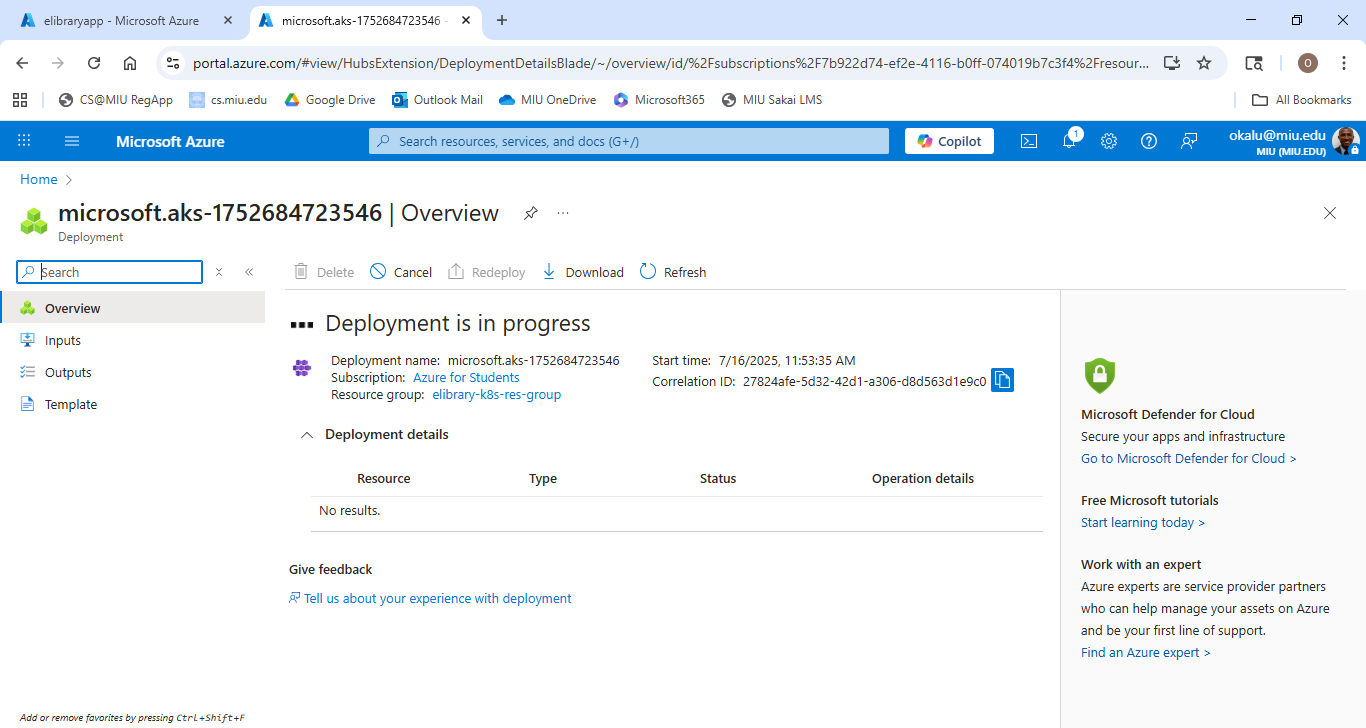
* 1. Sign-in to your Azure account. Execute > az login

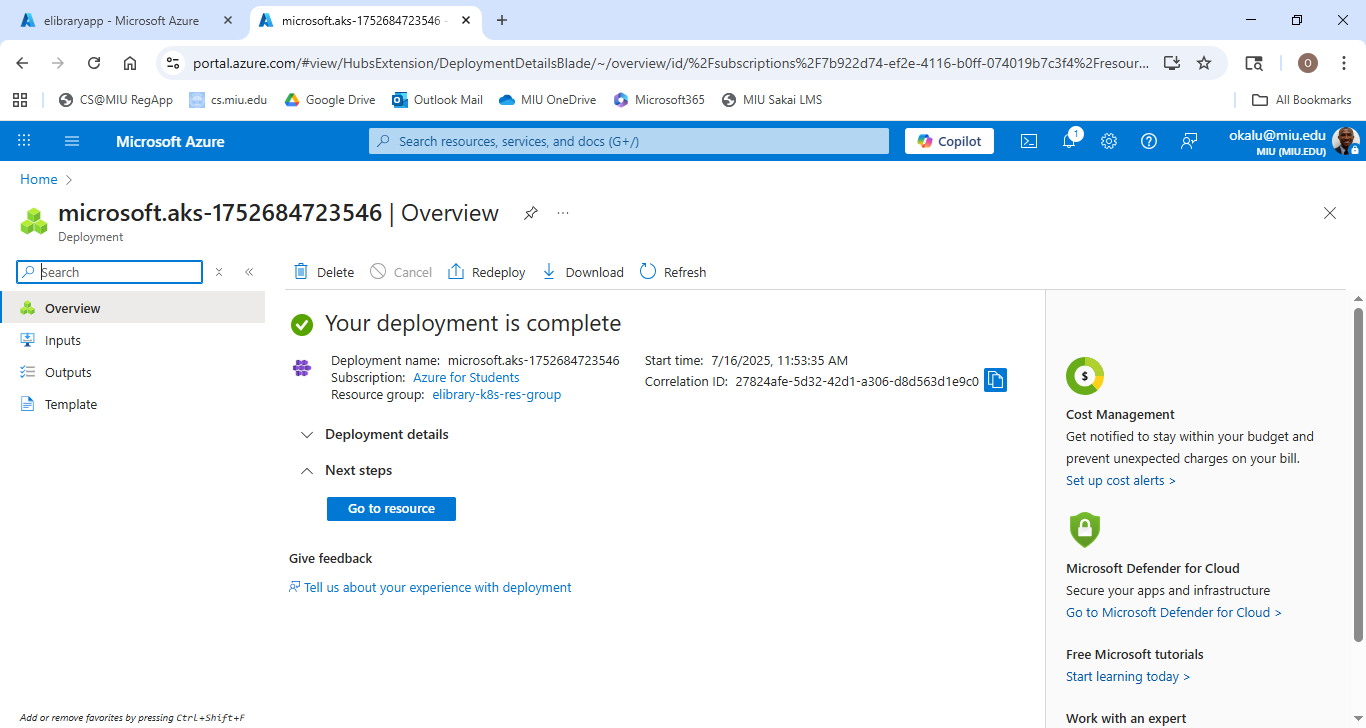
1. Go to the Azure portal – <https://portal.azure.com> and Search/Select/Open Kubernetes service and Create a new Kubernetes cluster.



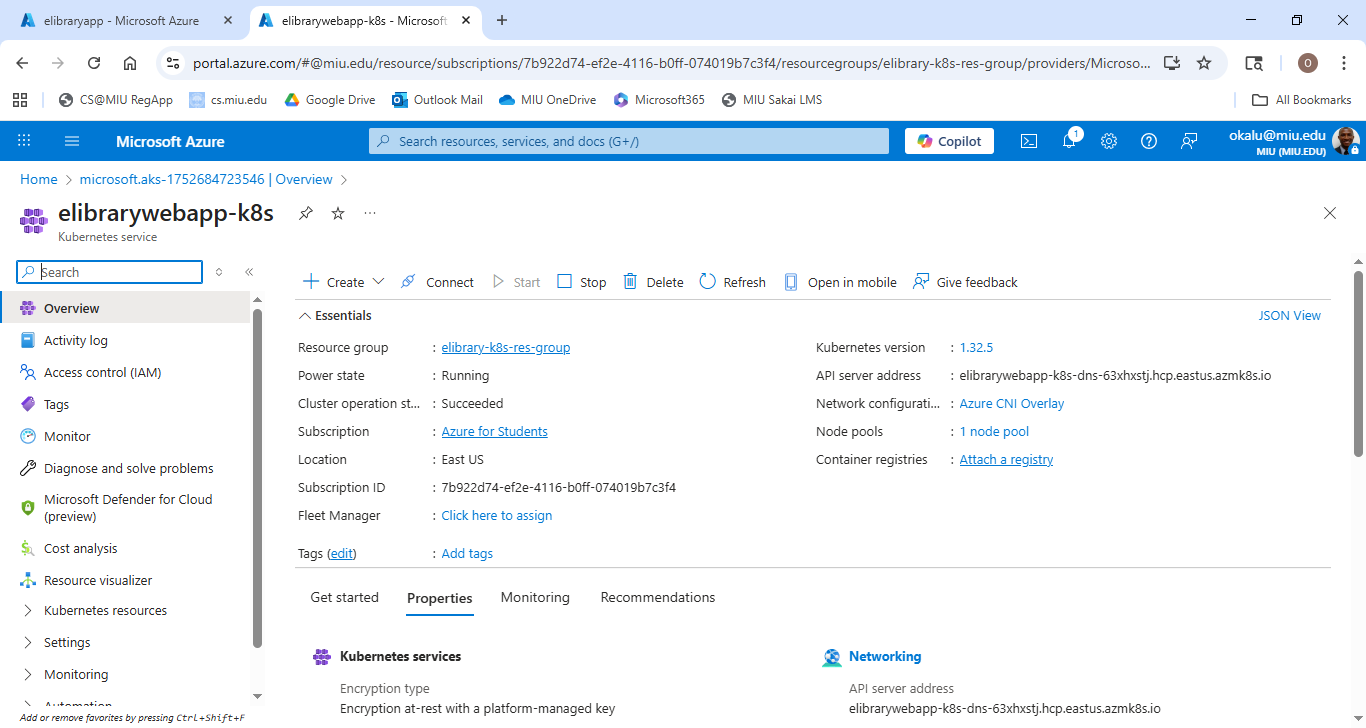


1. Go through the Deployment Creation wizard

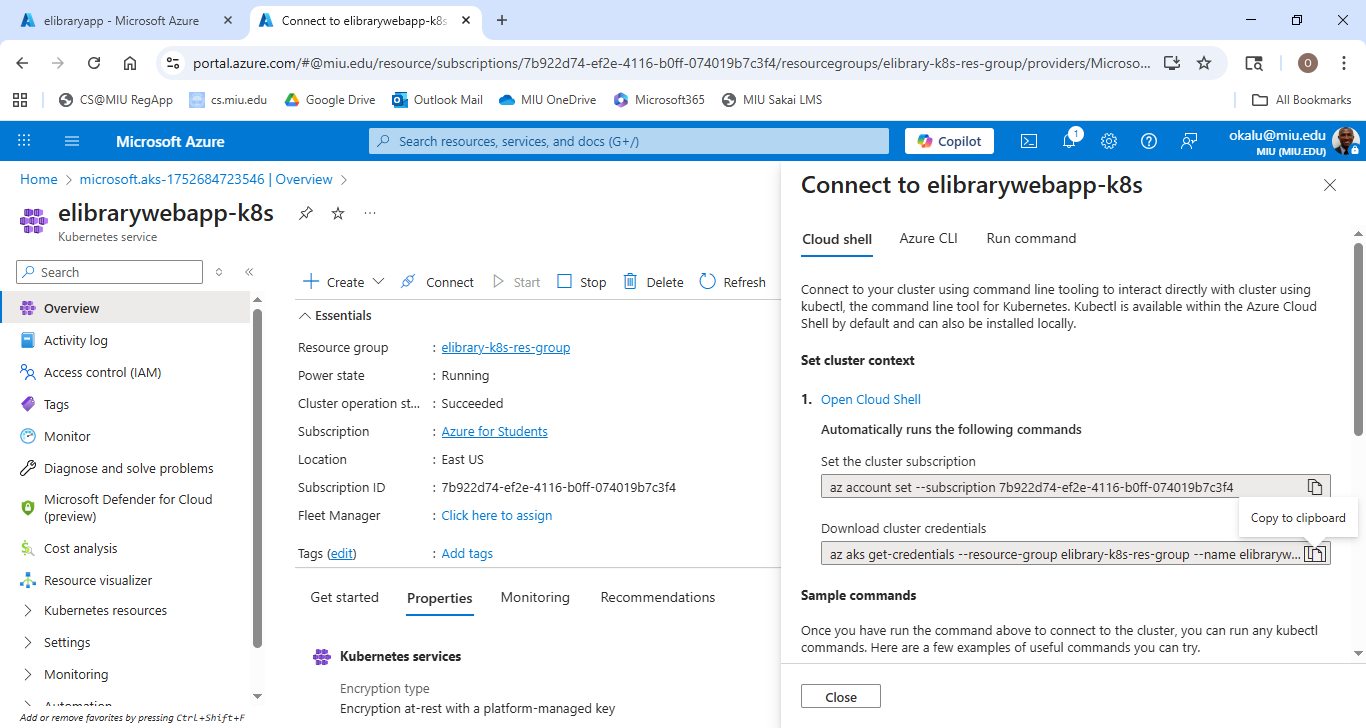




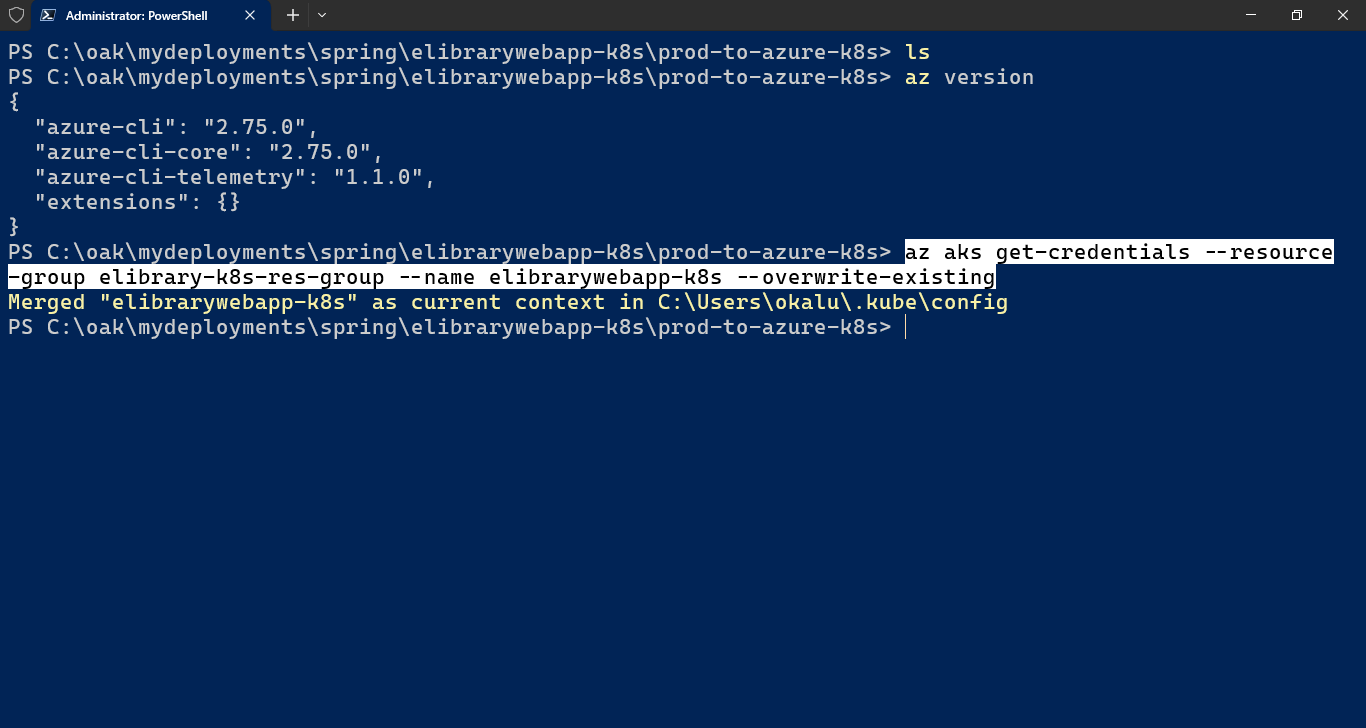
1. Click “Go to resource”



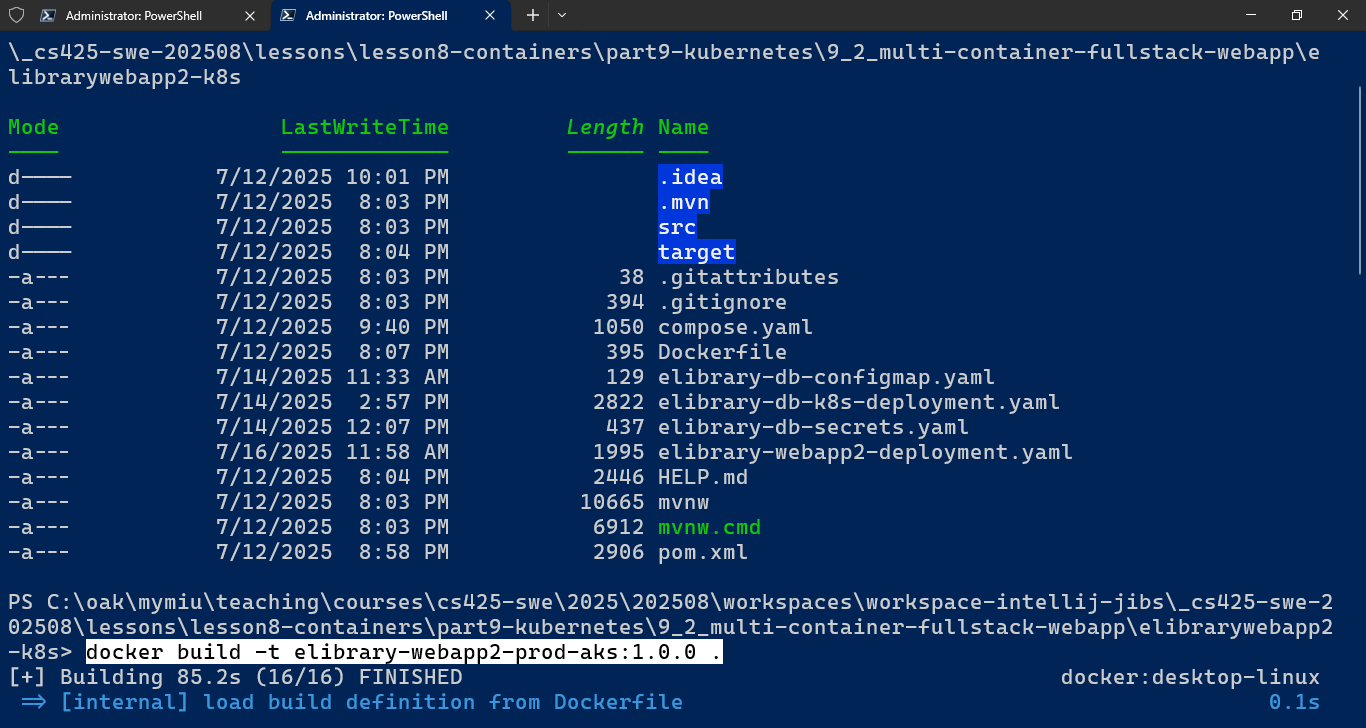
1. Click on the “Connect” tab and copy the cmd string given in “Download Cluster Credentials” textfield

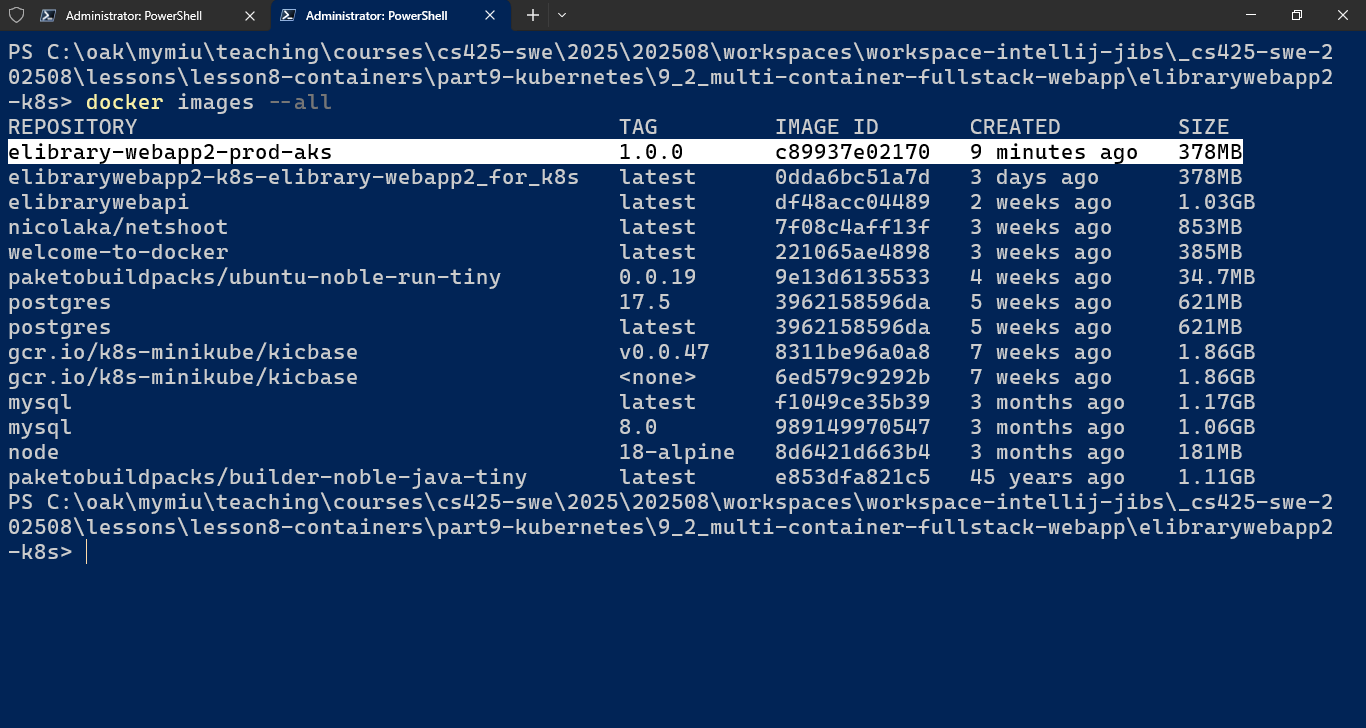


1. Open a Terminal/Cmd Shell and execute the cmd

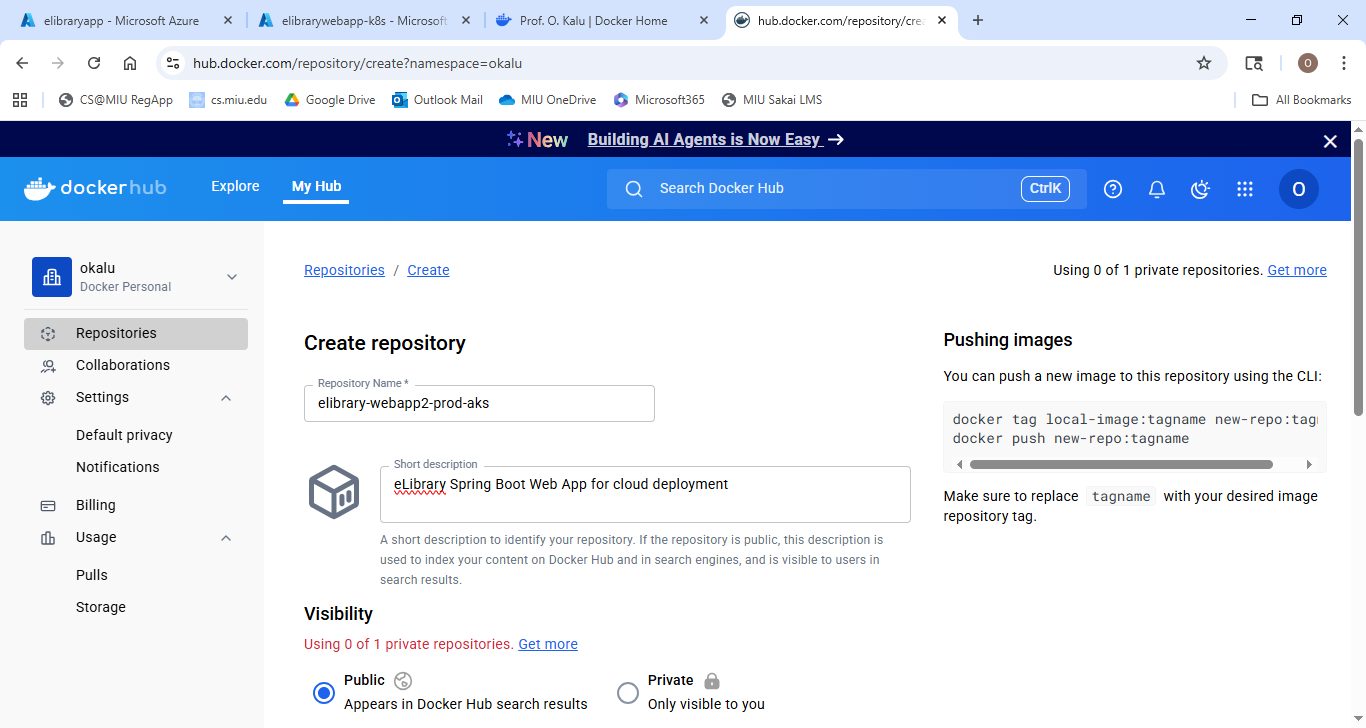
> 

1. Build a docker image of the webapp

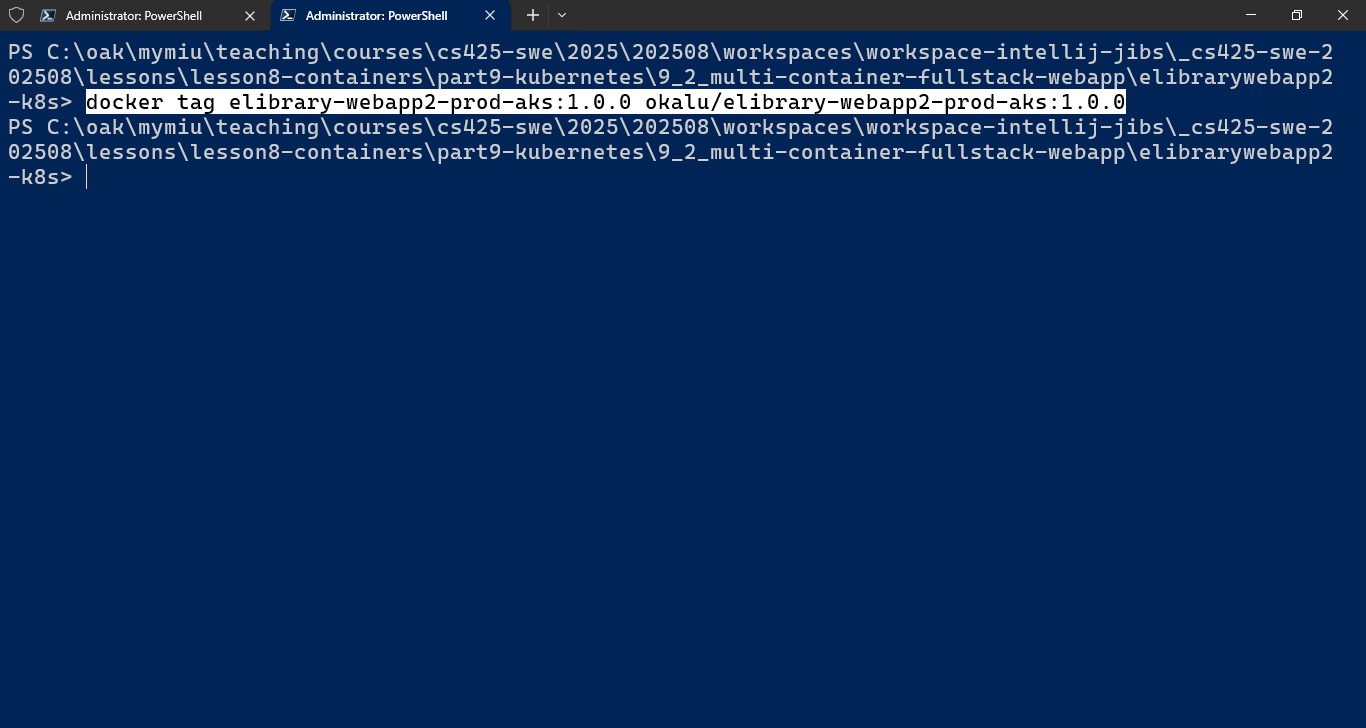


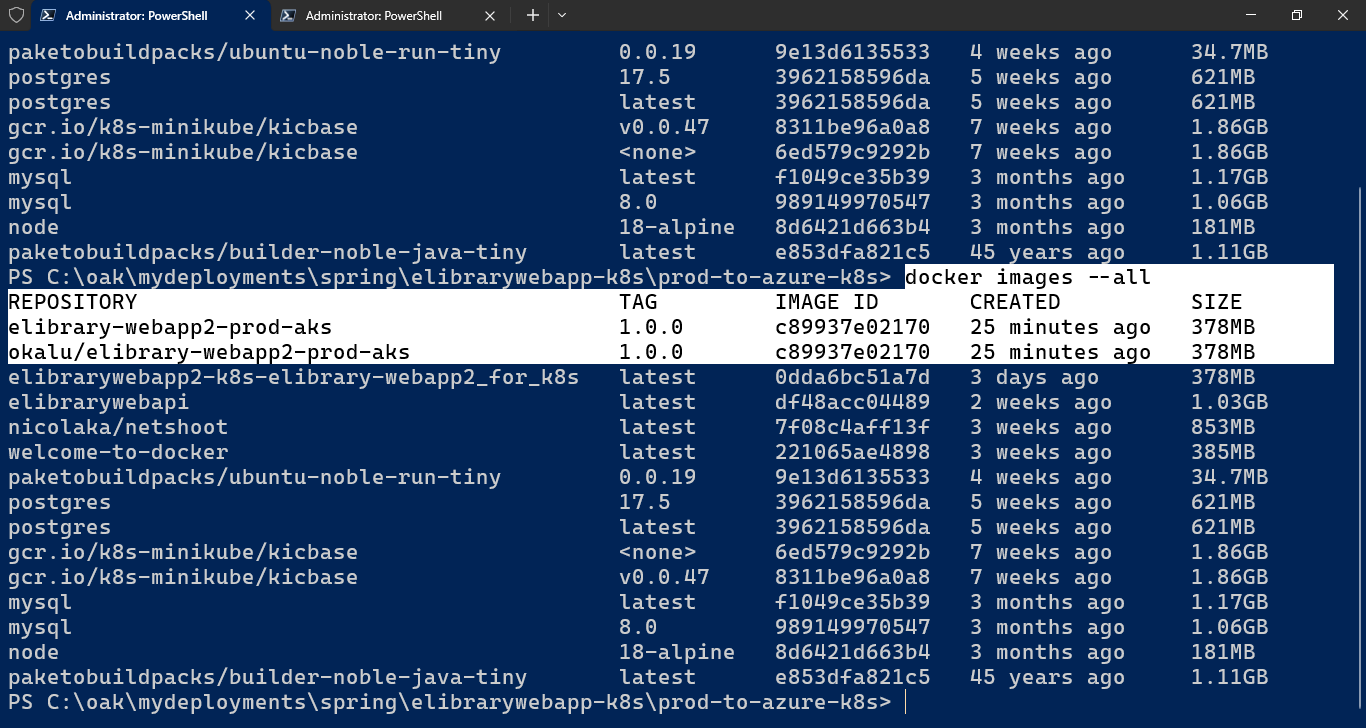


1. Push the image to a Docker Registry (such as your Docker hub OR Azure Container Registry). For Docker, Create a new Repository in your Docker hub:

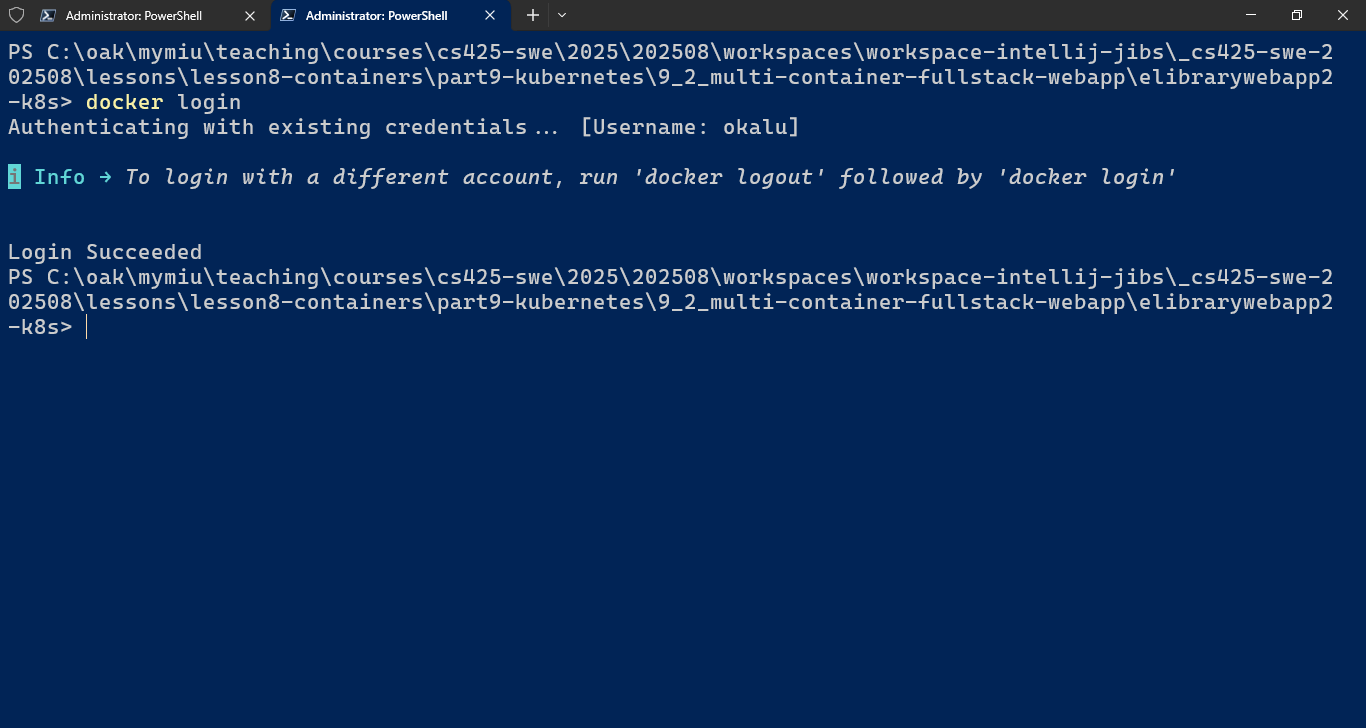


Rename the new image:

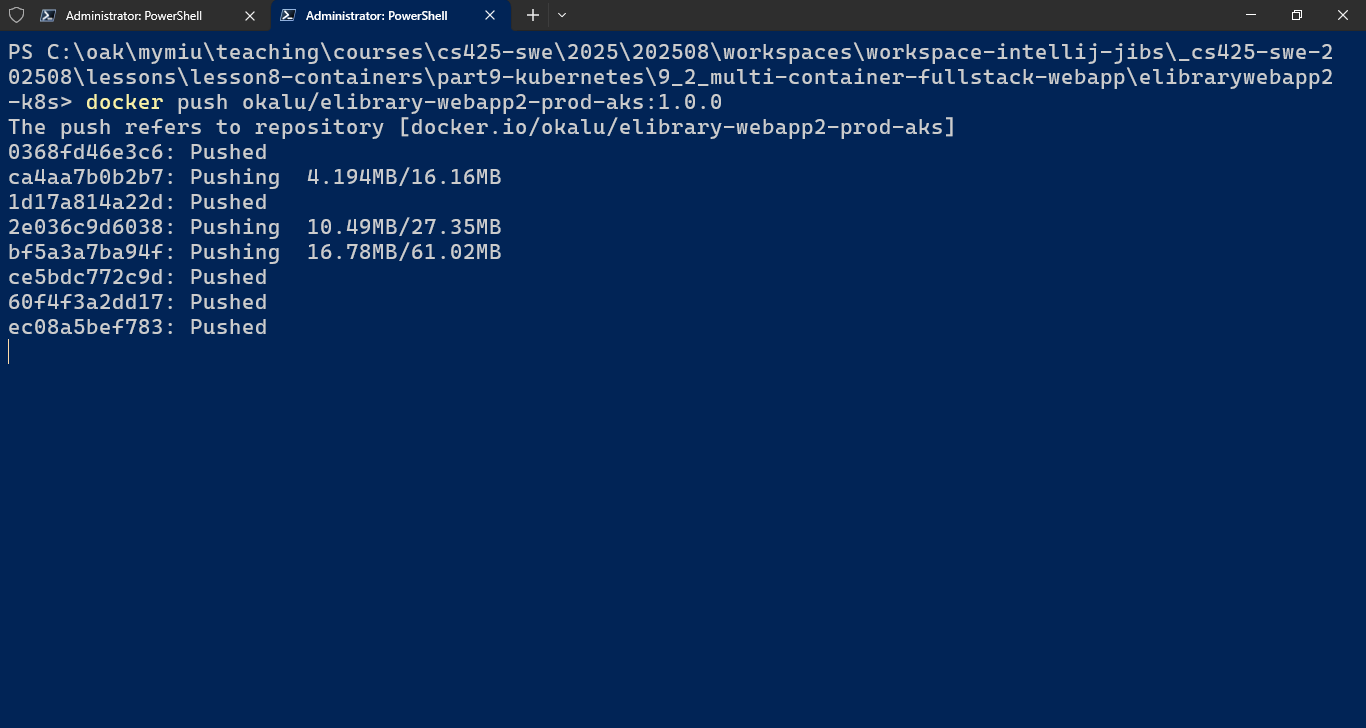


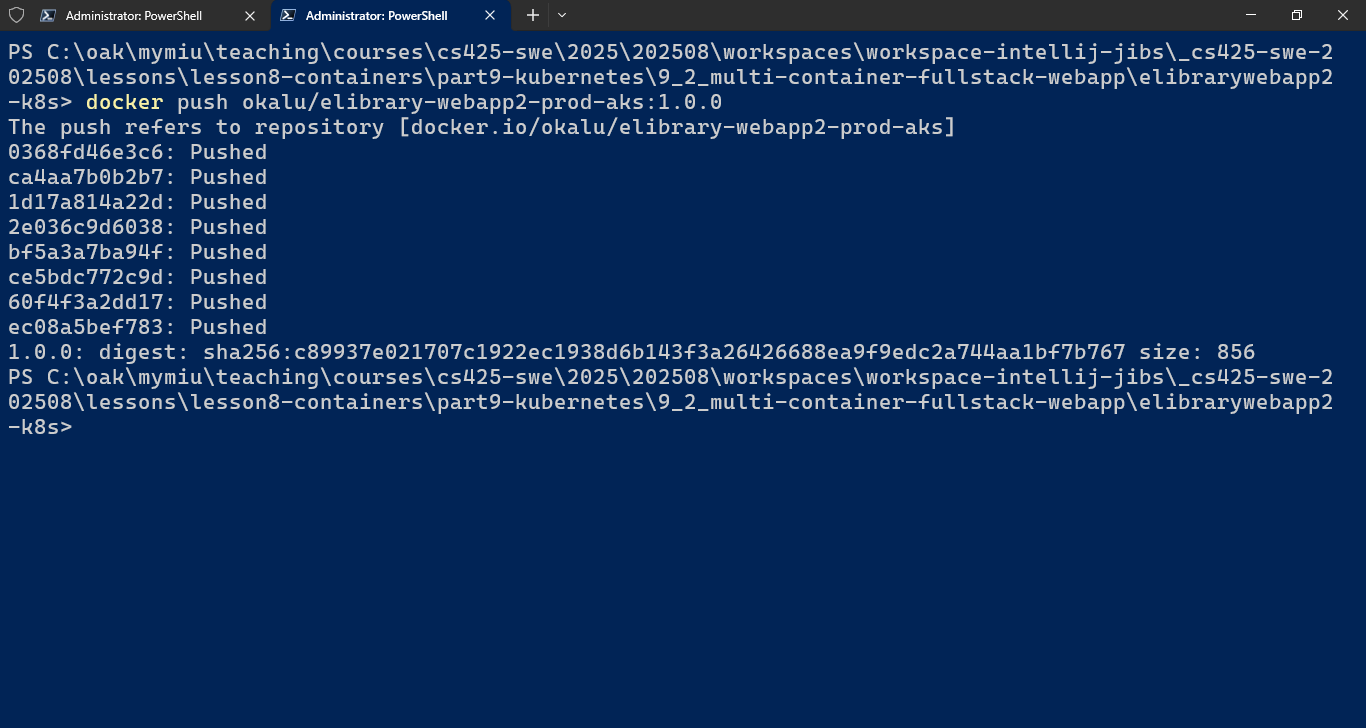


Execute ‘docker login’ on the CLI

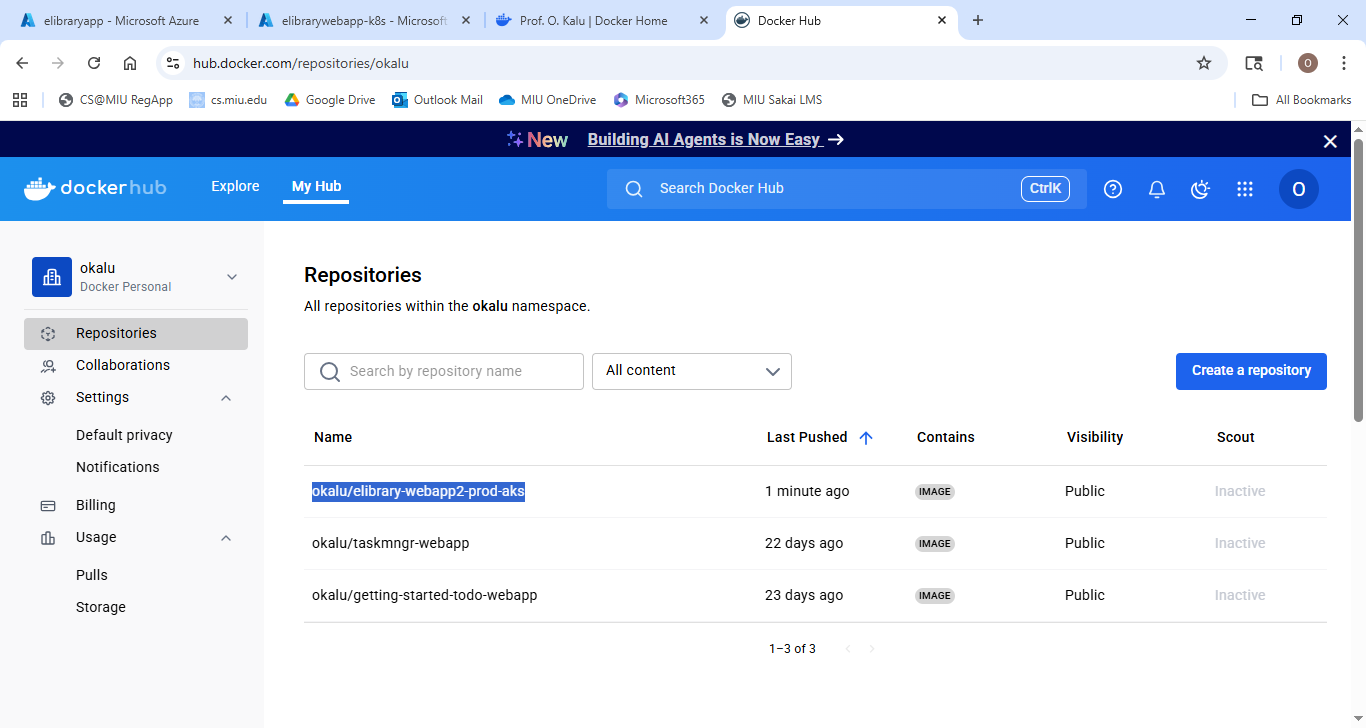


Push the new image to the repository using the docker push cmd:

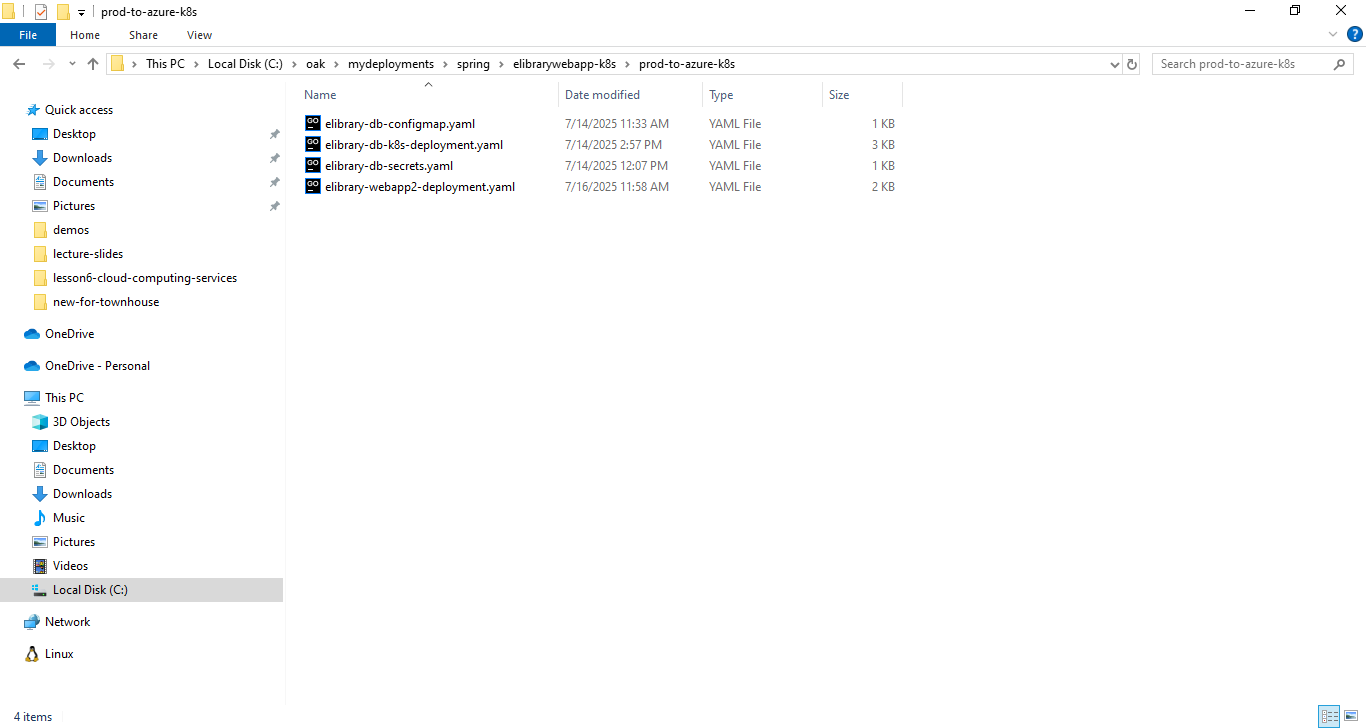




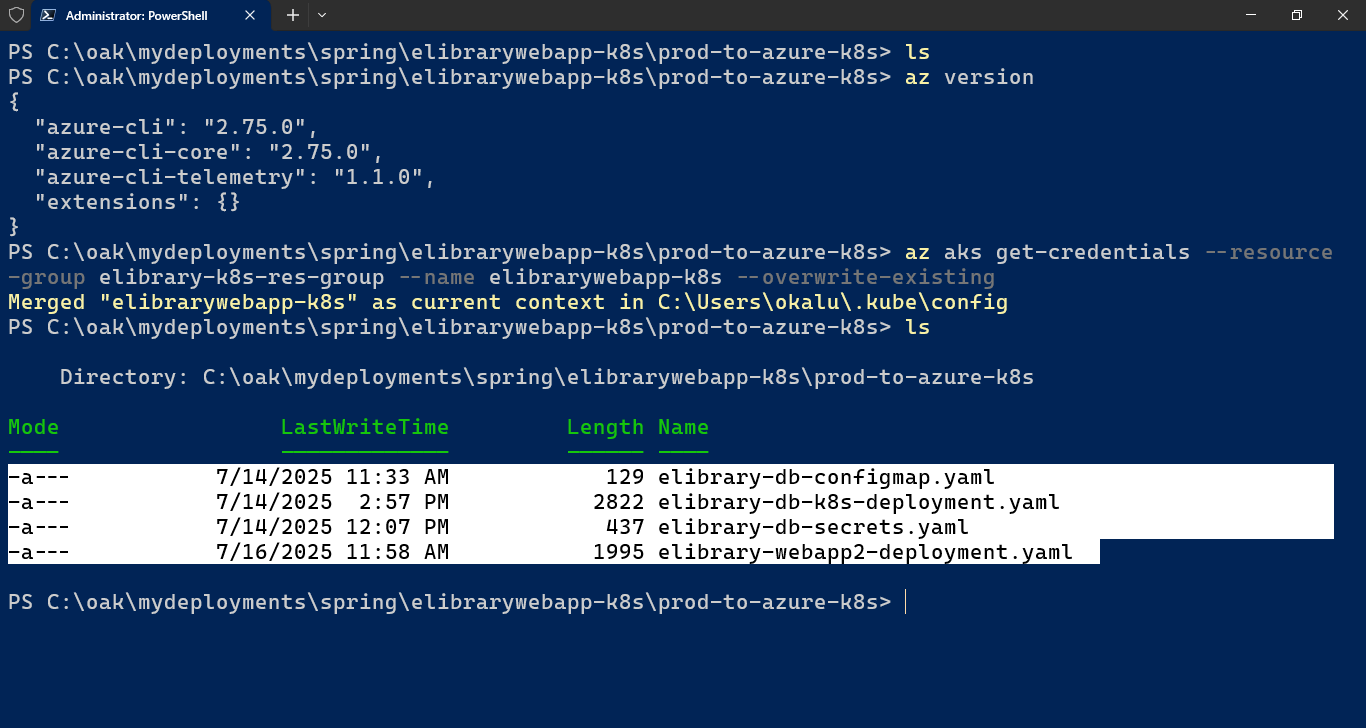
Check that the new image is listed in your Docker hub



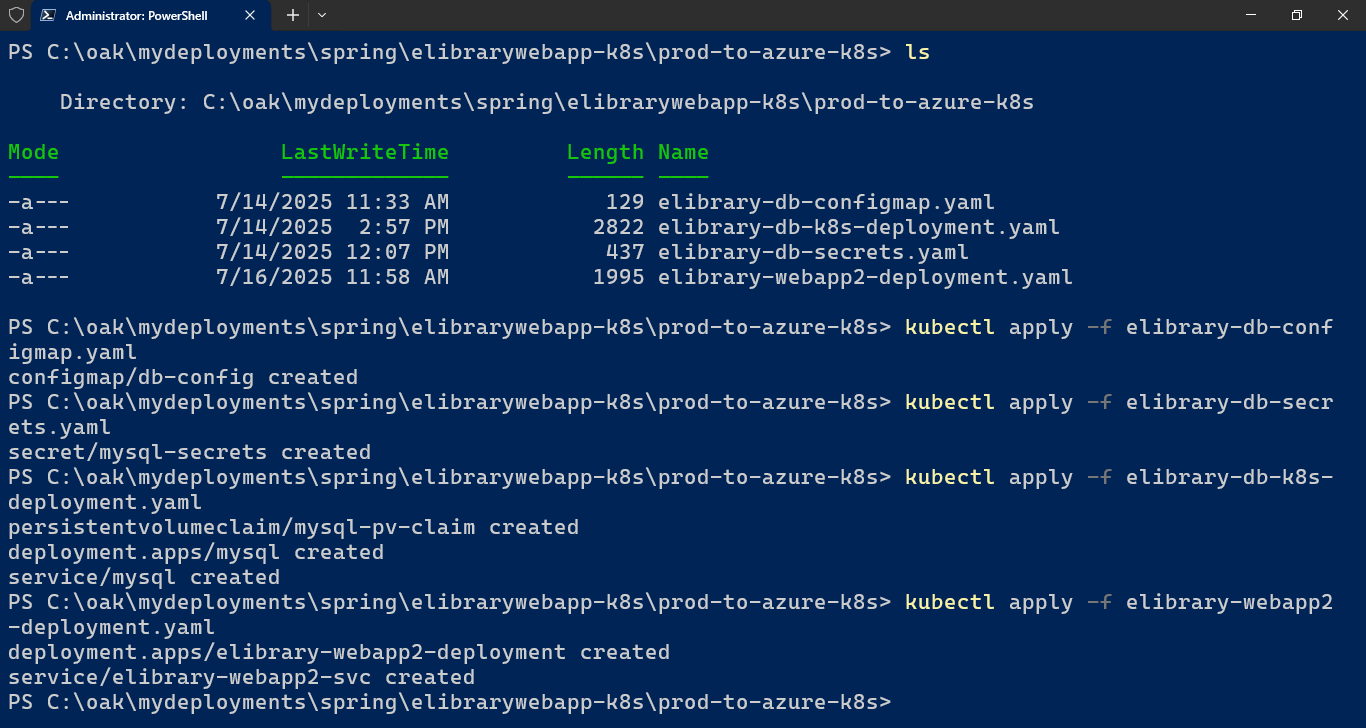
1. Copy the k8s \*.yaml file(s) to a suitable folder



And cd into there



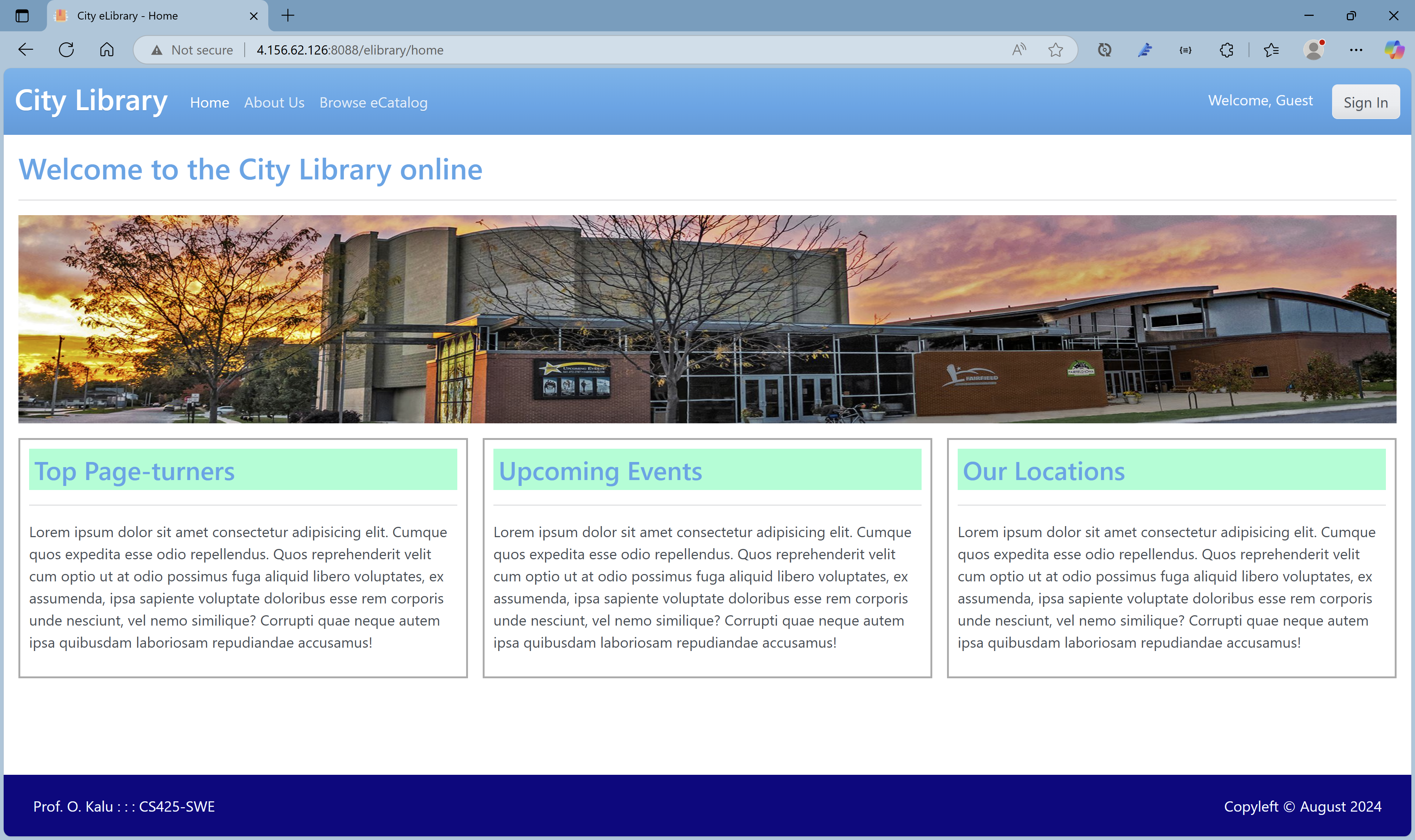
1. Execute ‘kubectl apply –f ...’ for each of the yaml files



1. Check that the services and the deployments and the pods are all up and running:



1. Access the webapp service using the external IP address and Port



1. ... The End!!!