Objectives

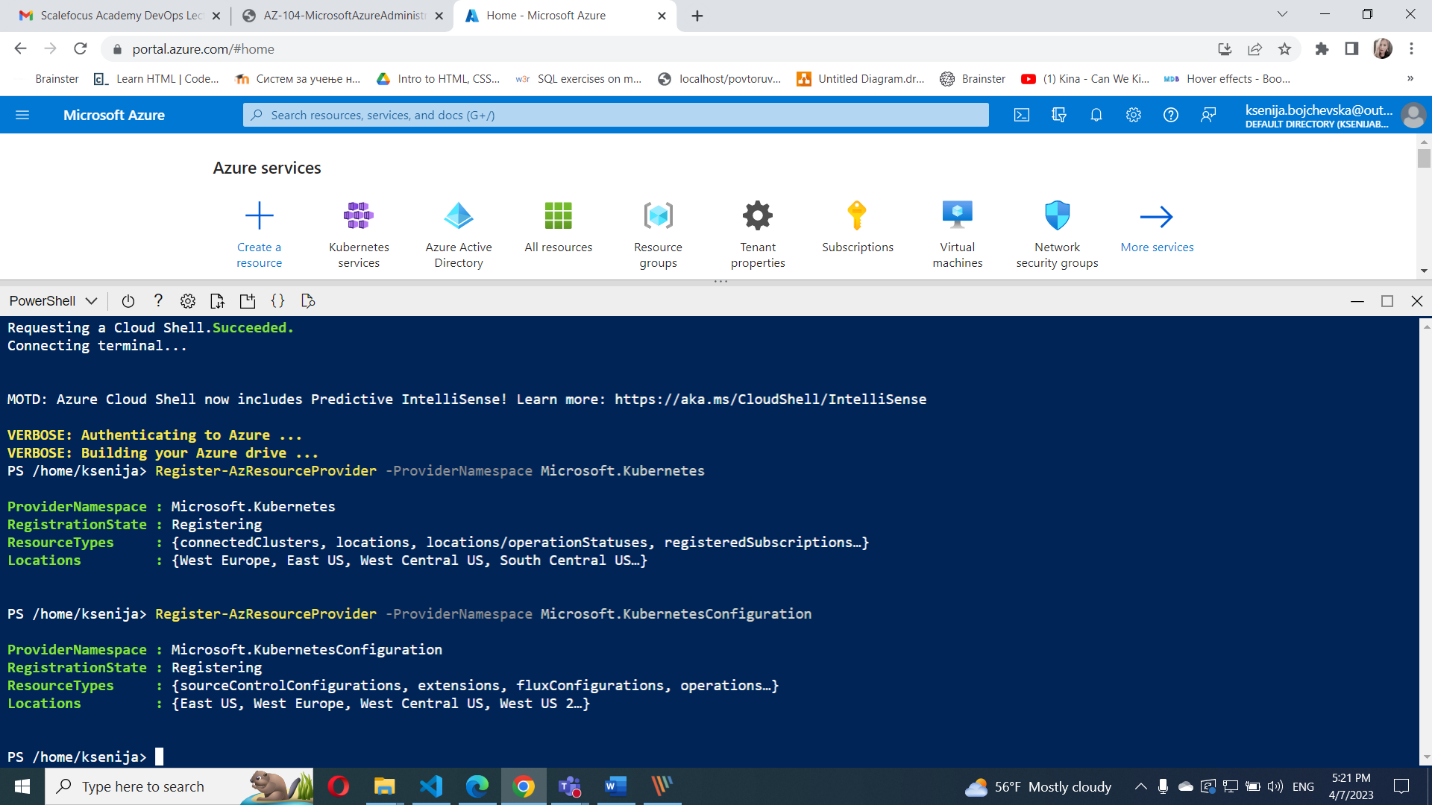
In this lab, you will:

* Task 1: Register the Microsoft.Kubernetes and Microsoft.KubernetesConfiguration resource providers.
* Task 2: Deploy an Azure Kubernetes Service cluster
* Task 3: Deploy pods into the Azure Kubernetes Service cluster
* Task 4: Scale containerized workloads in the Azure Kubernetes service cluster

From the Cloud Shell pane, run the following to register the Microsoft.Kubernetes and Microsoft.KubernetesConfiguration resource providers.

Register-AzResourceProvider -ProviderNamespace Microsoft.Kubernetes

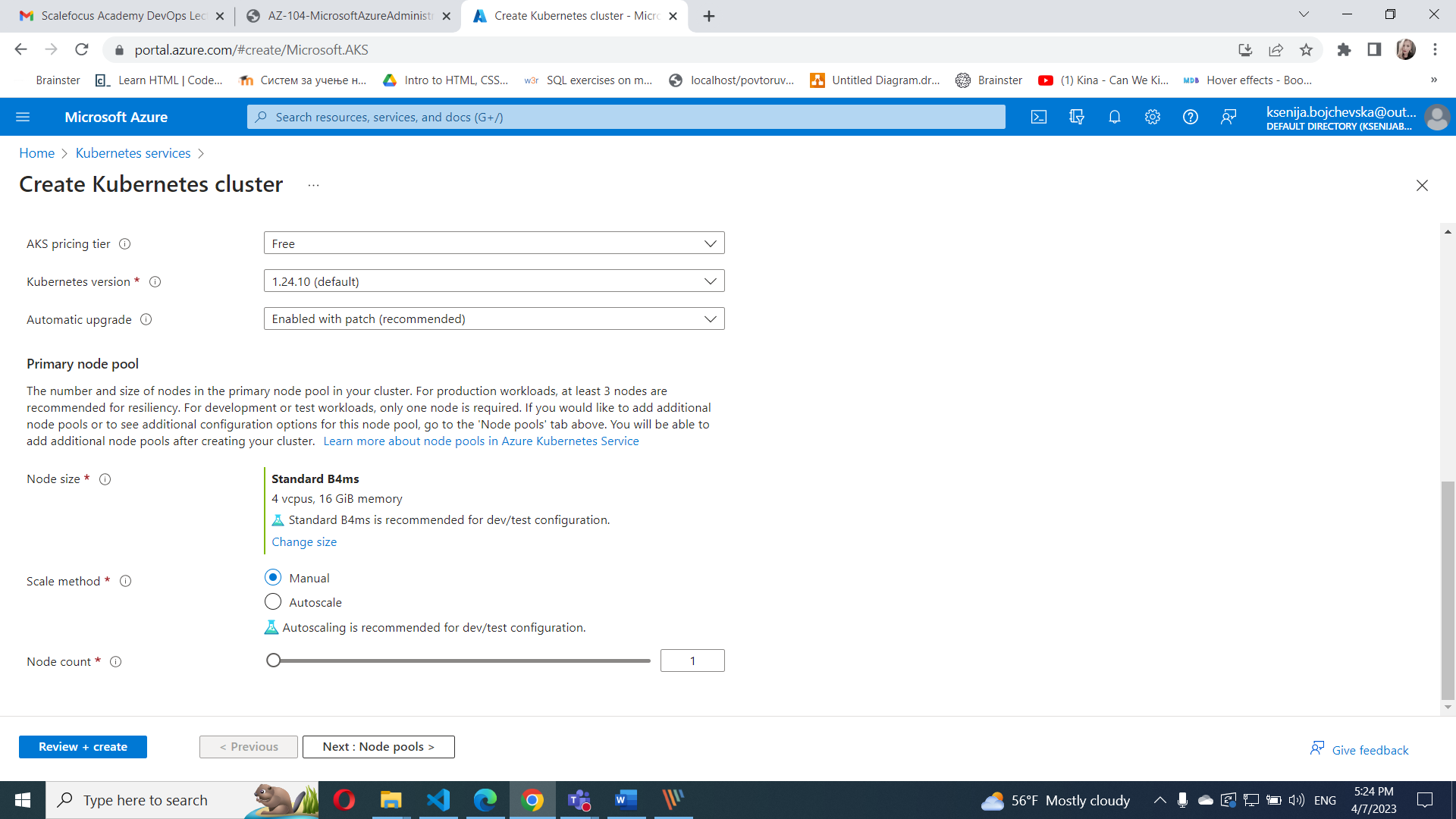
Register-AzResourceProvider -ProviderNamespace Microsoft.KubernetesConfiguration

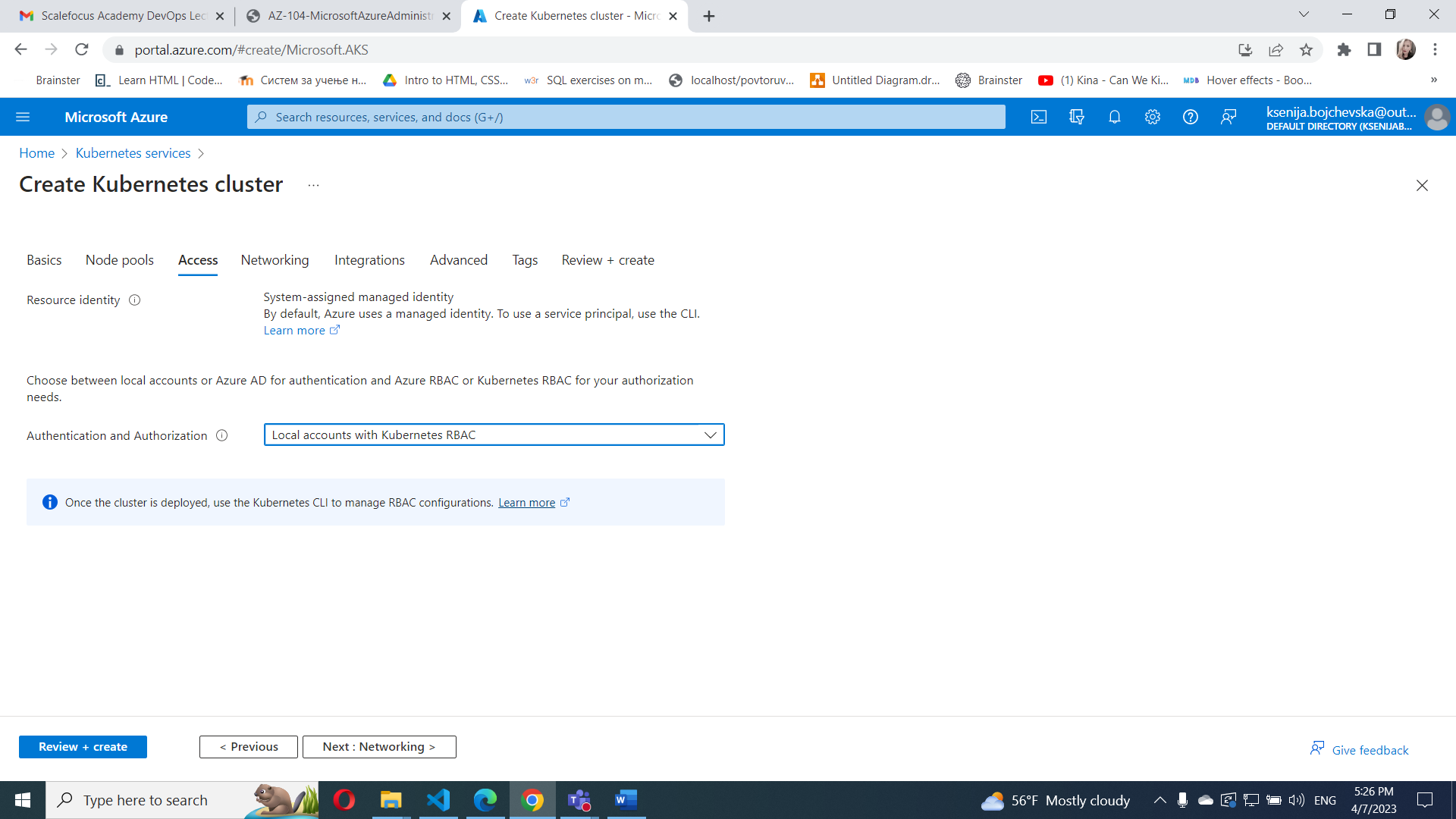
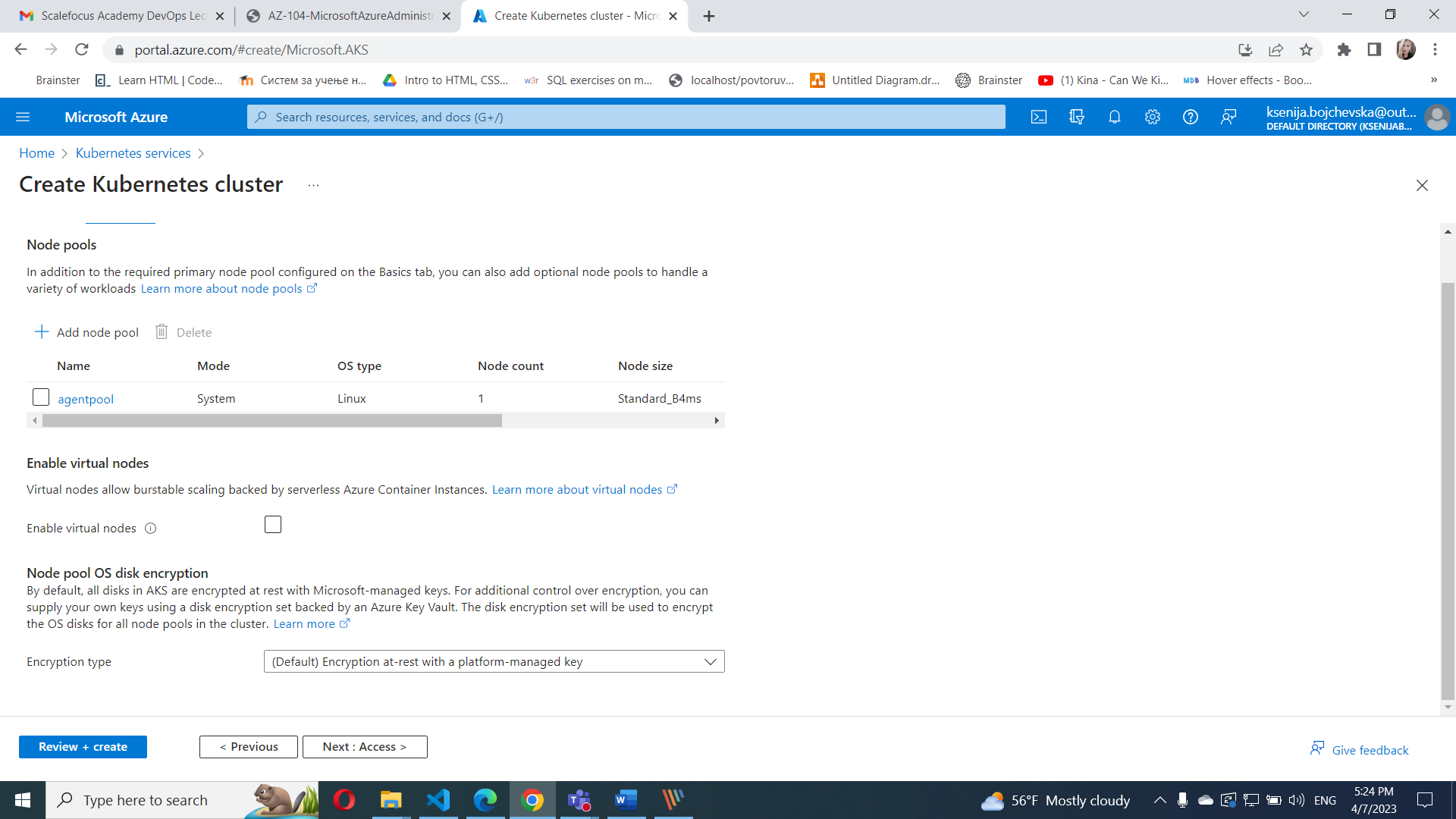


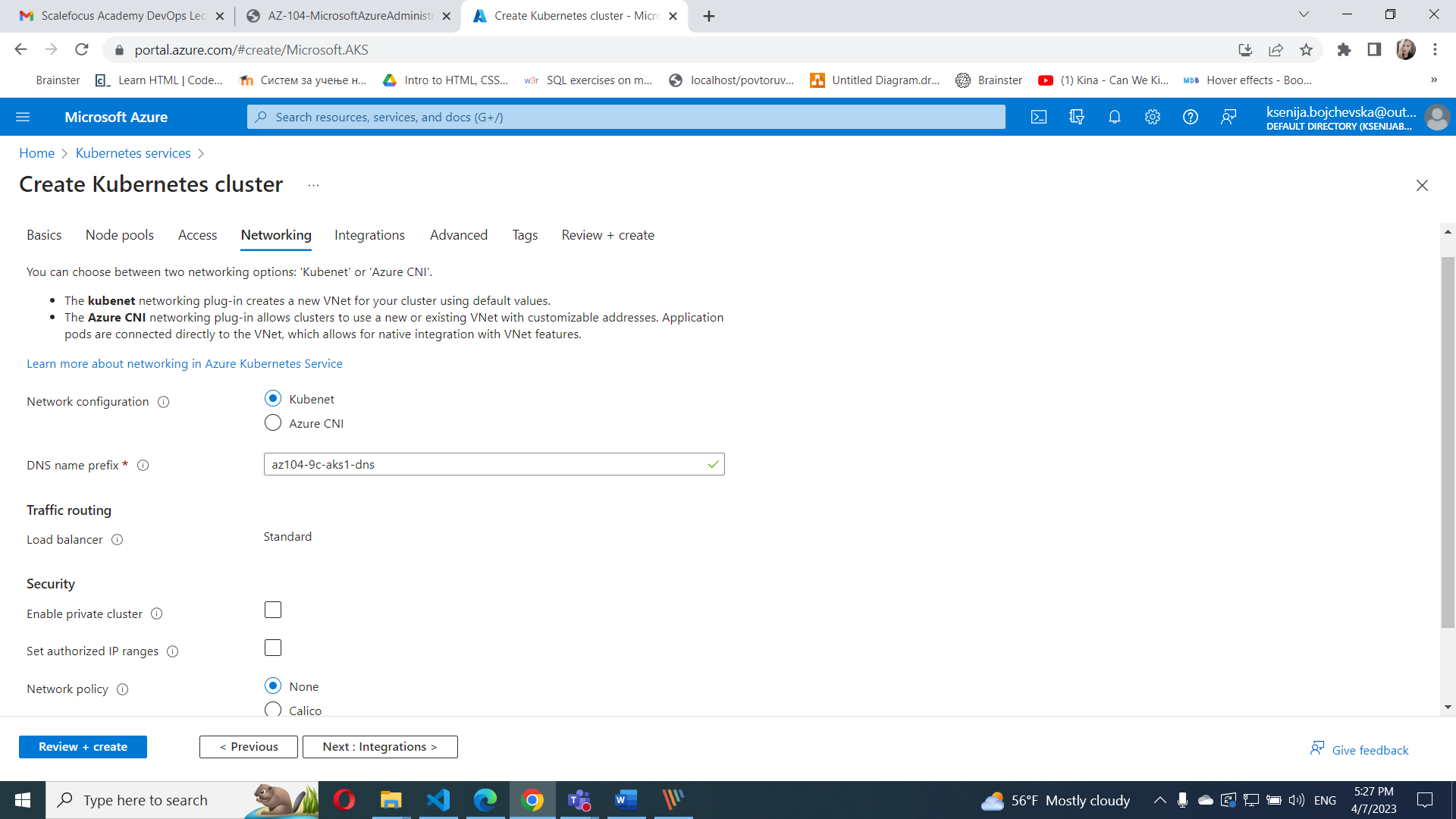
#### **Task 2: Deploy an Azure Kubernetes Service cluster**

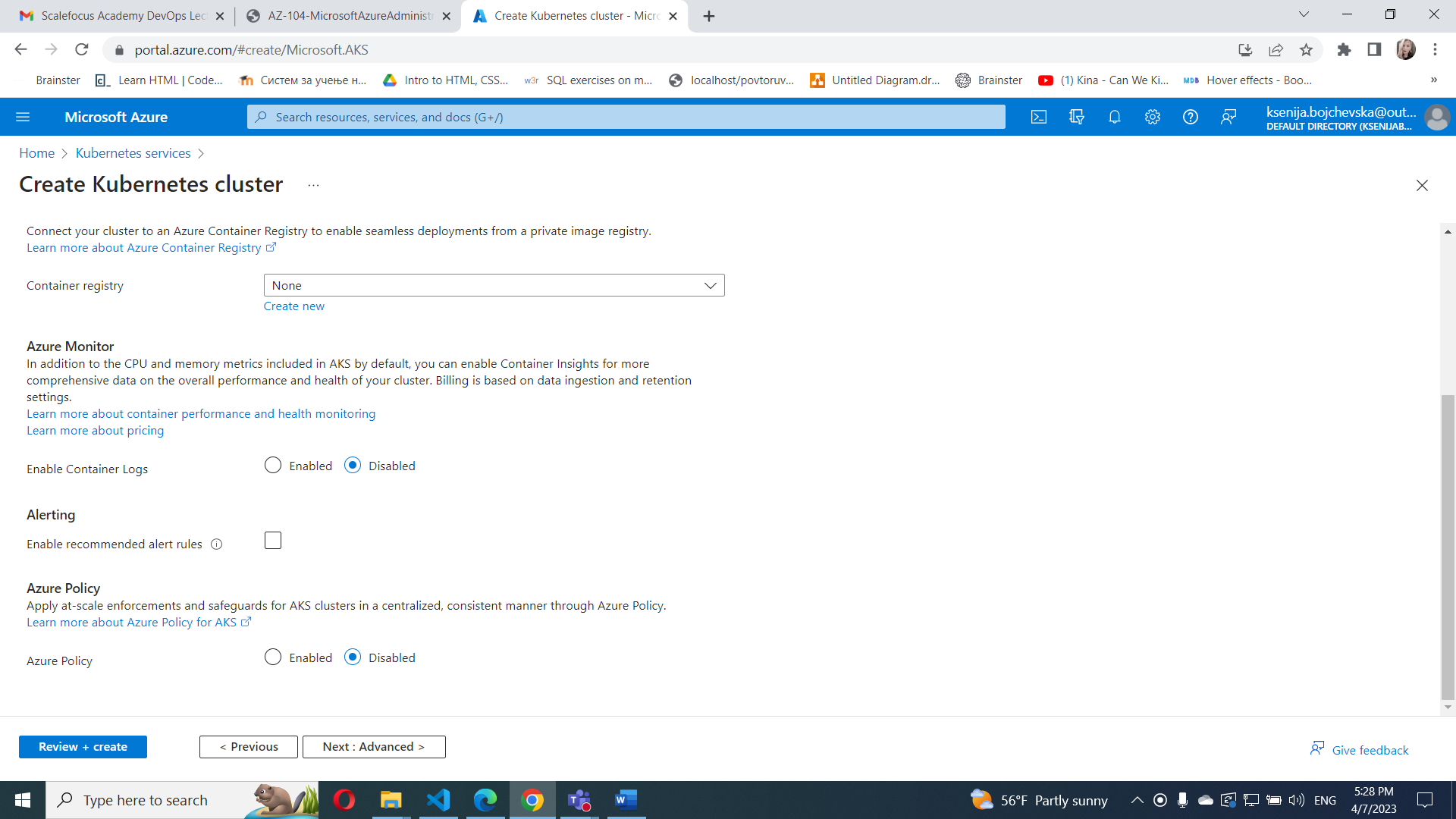
1. In the Azure portal, search for locate **Kubernetes services** and then, on the **Kubernetes services** blade, click **+ Create**, and then click **+ Create a Kubernetes cluster**.
2. On the **Basics** tab of the **Create Kubernetes cluster** blade, specify the following settings (leave others with their default values):





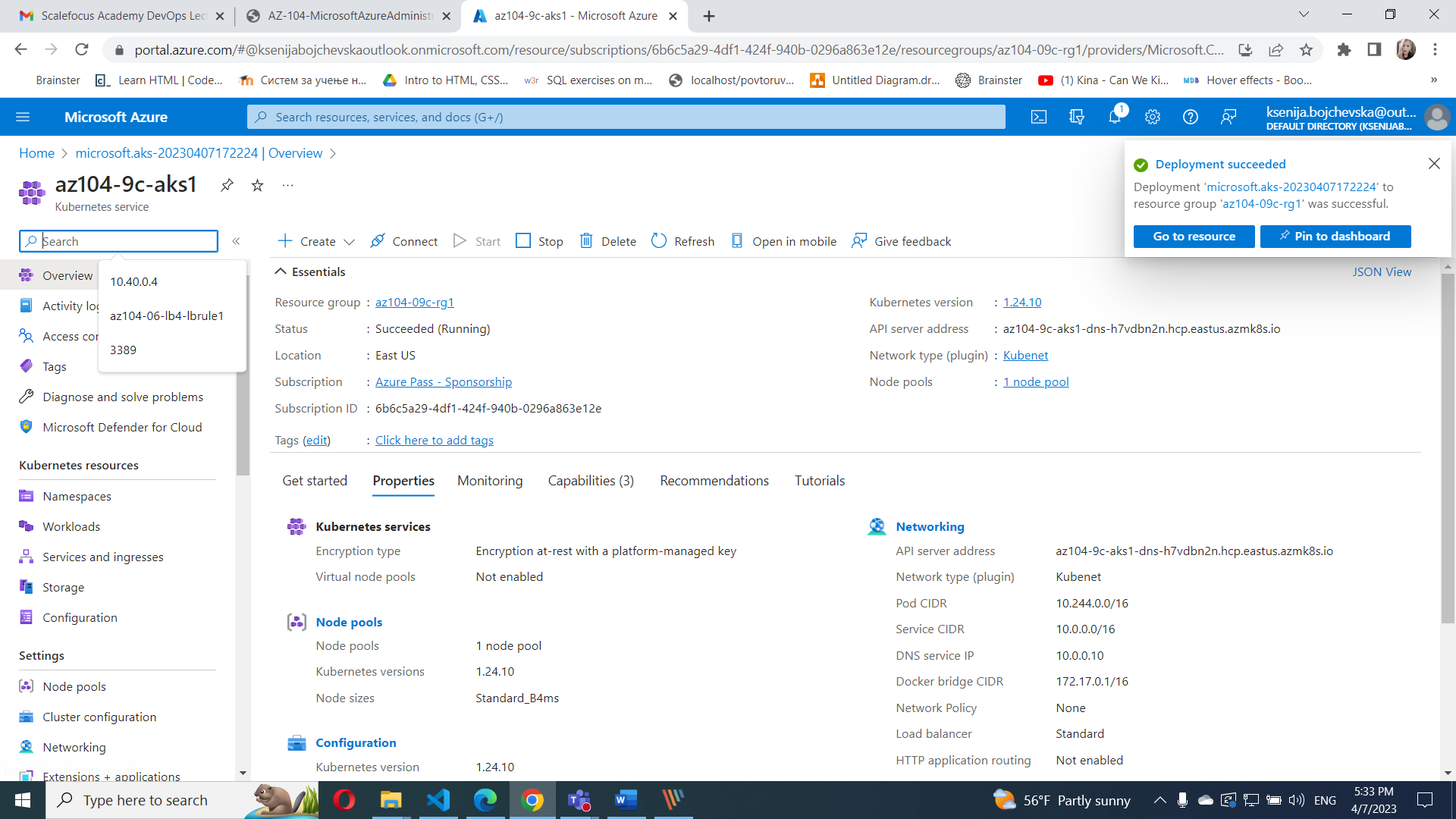


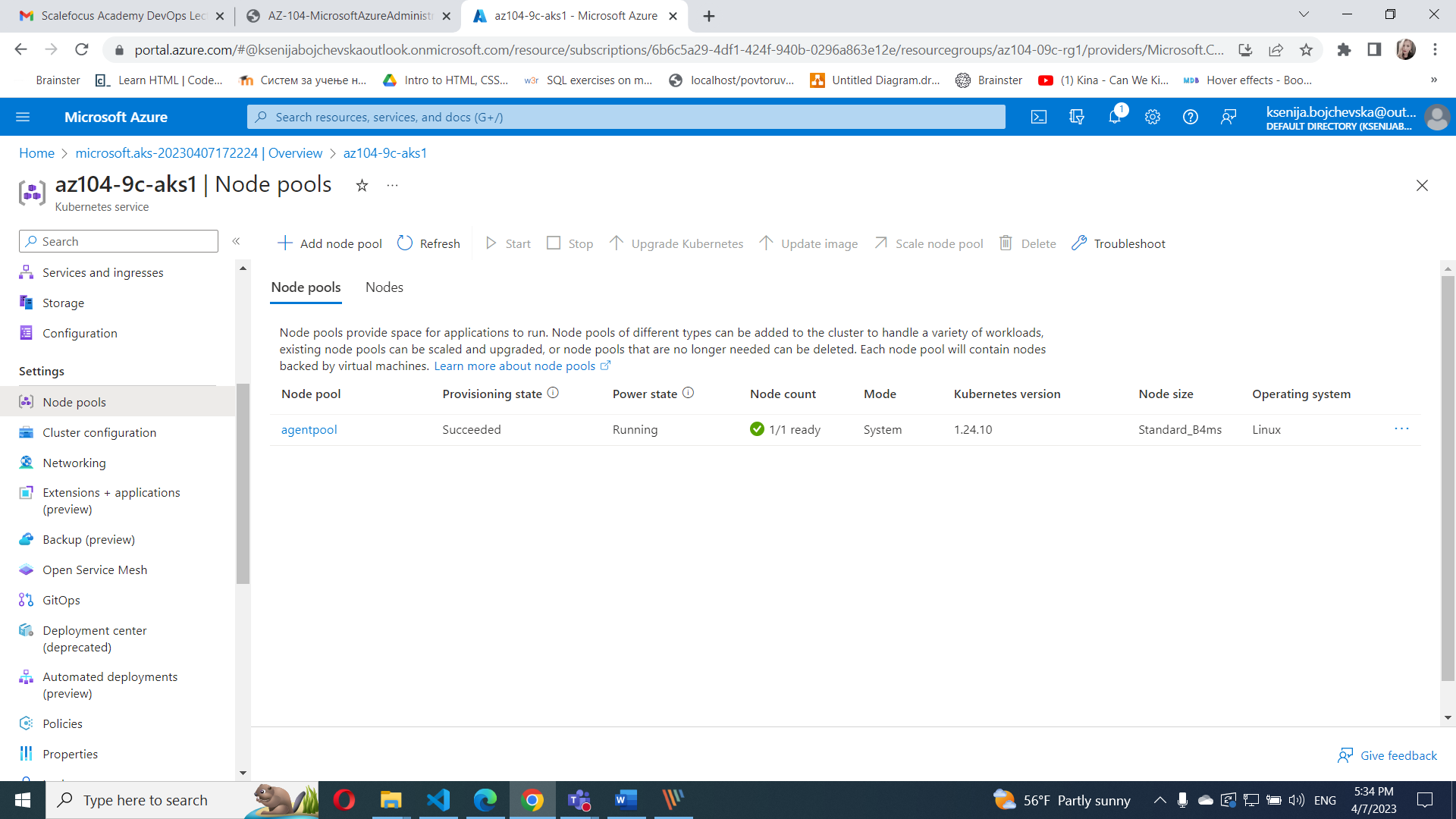




#### **Task 3: Deploy pods into the Azure Kubernetes Service cluster**

1. On the deployment blade, click the **Go to resource** link.



1. On the **az104-9c-aks1** Kubernetes service blade, in the **Settings** section, click **Node pools**.
2. On the **az104-9c-aks1 - Node pools** blade, verify that the cluster consists of a single pool with one node.
3. 
4. In the Azure portal, open the **Azure Cloud Shell** by clicking on the icon in the top right of the Azure Portal.
5. Switch the **Azure Cloud Shell** to **Bash** (black background).

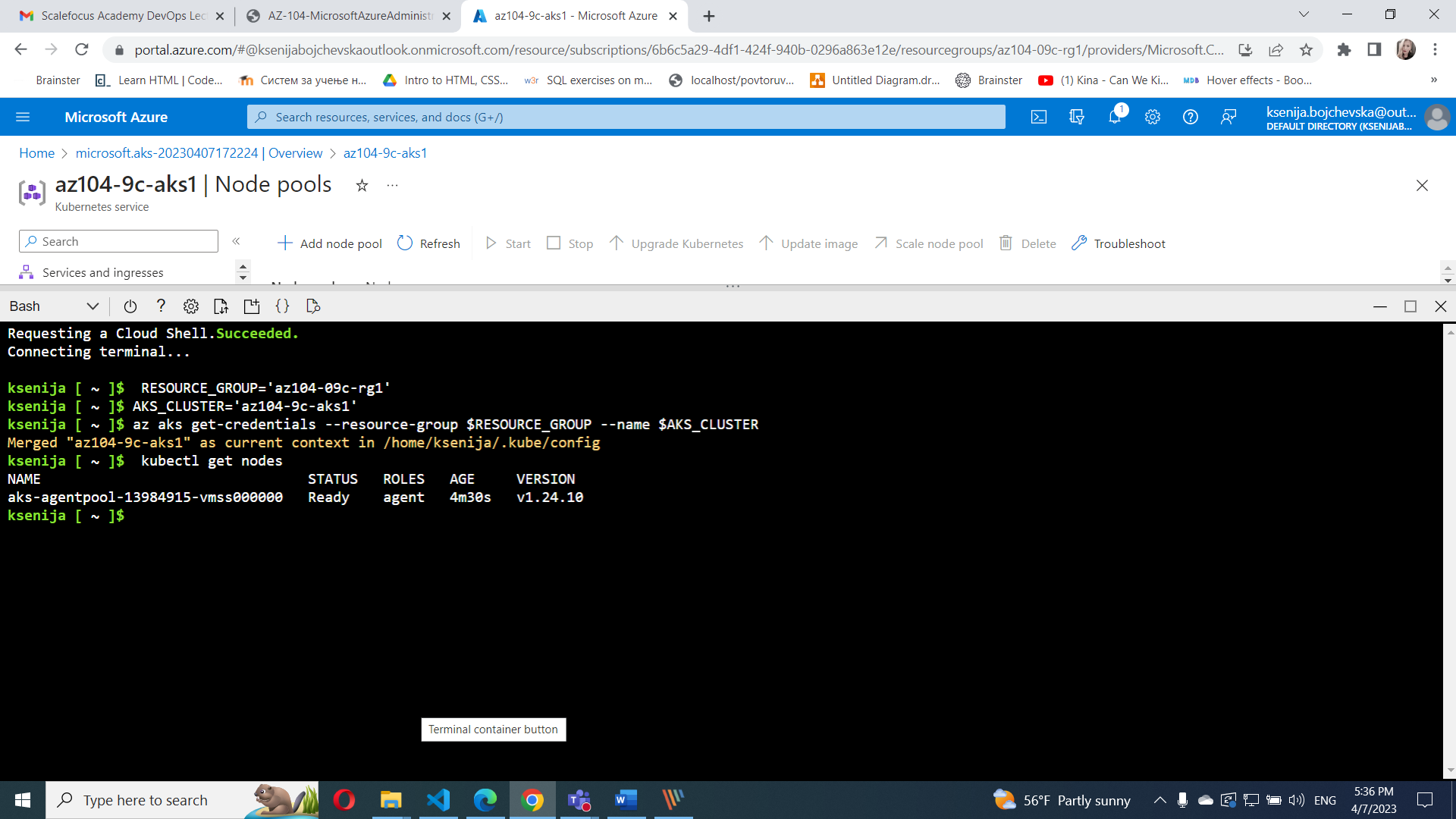
From the Cloud Shell pane, run the following to retrieve the credentials to access the AKS cluster:

RESOURCE\_GROUP='az104-09c-rg1'

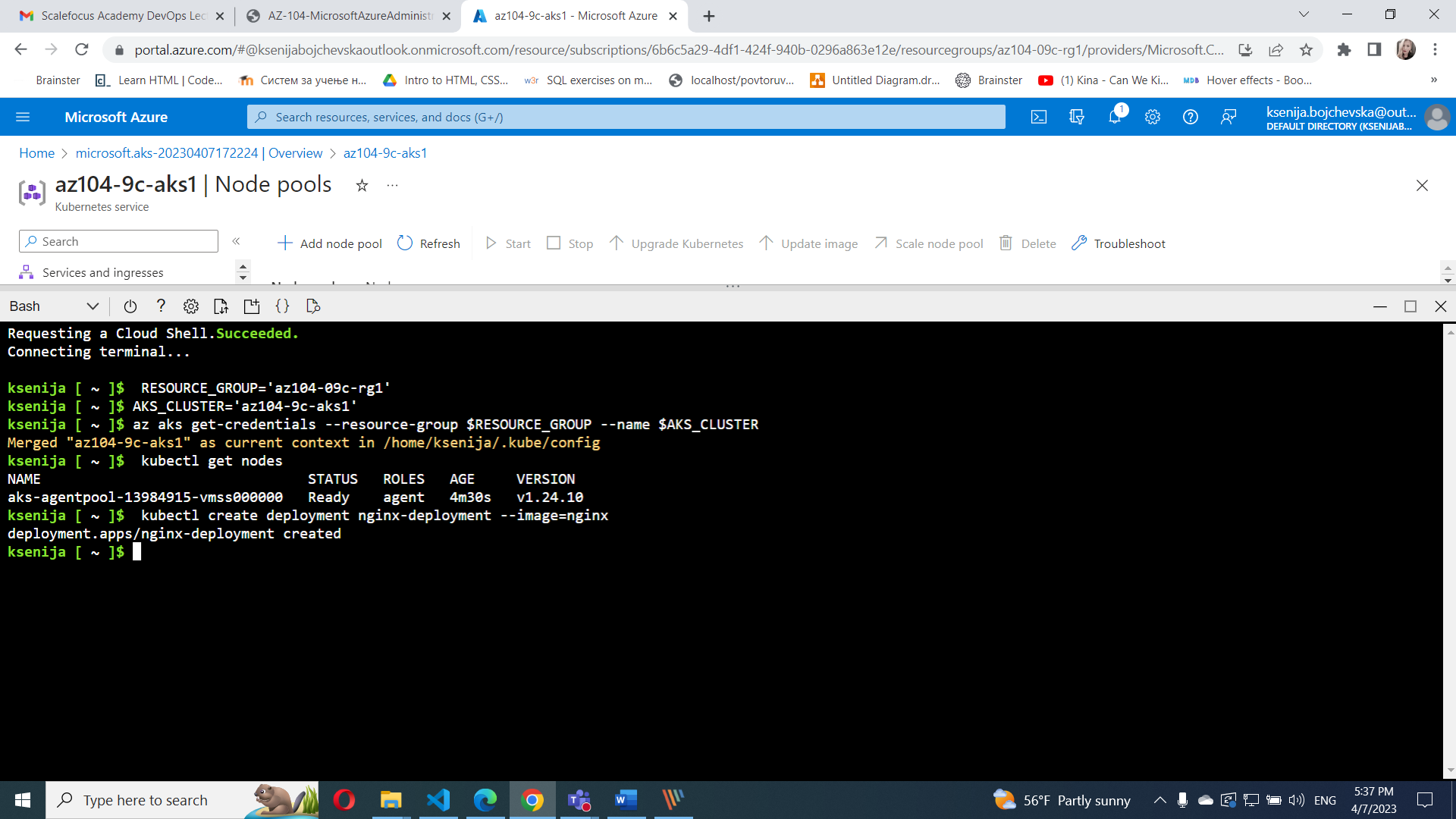
AKS\_CLUSTER='az104-9c-aks1'

az aks get-credentials --resource-group $RESOURCE\_GROUP --name $AKS\_CLUSTER

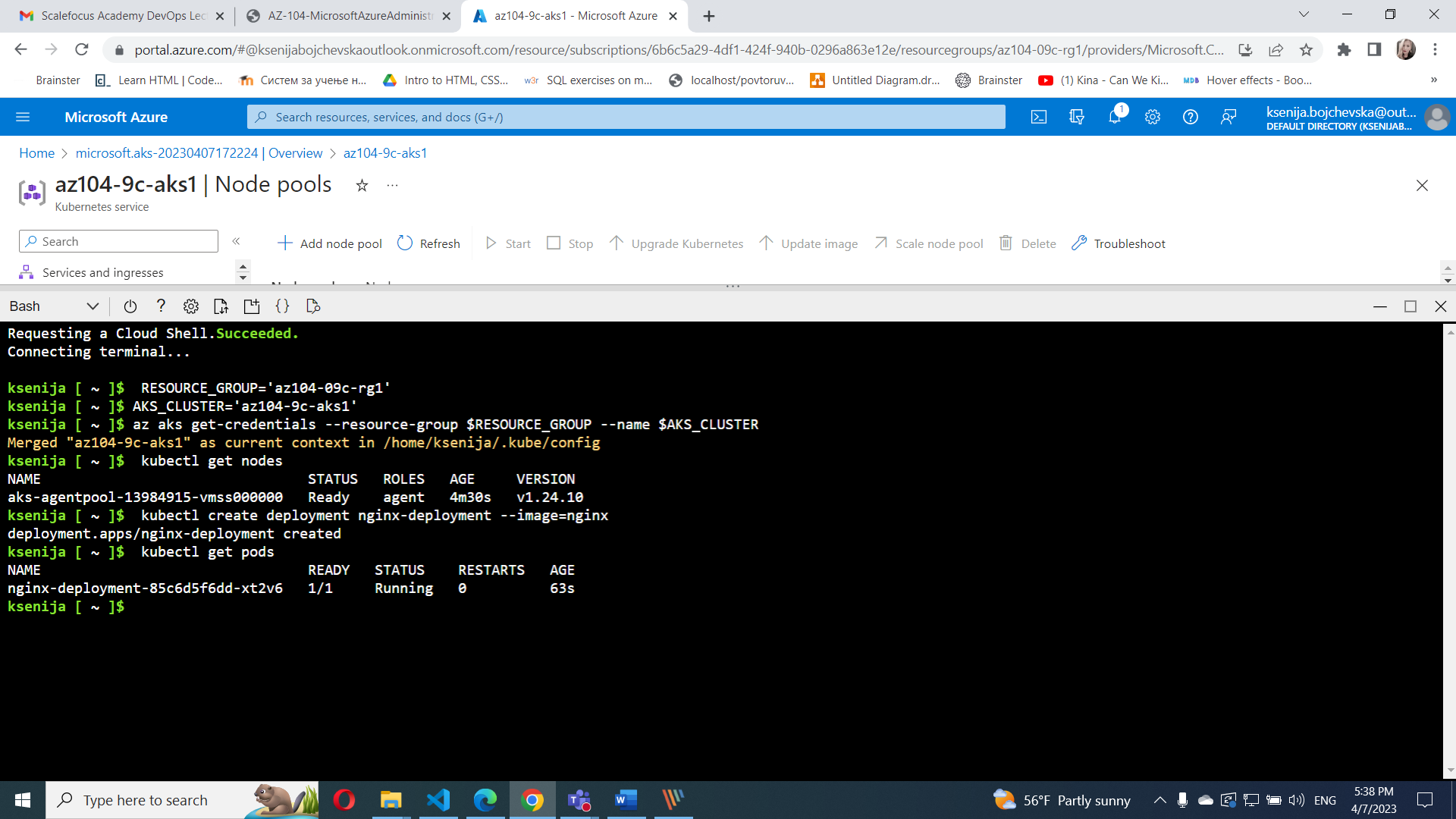
kubectl get nodes



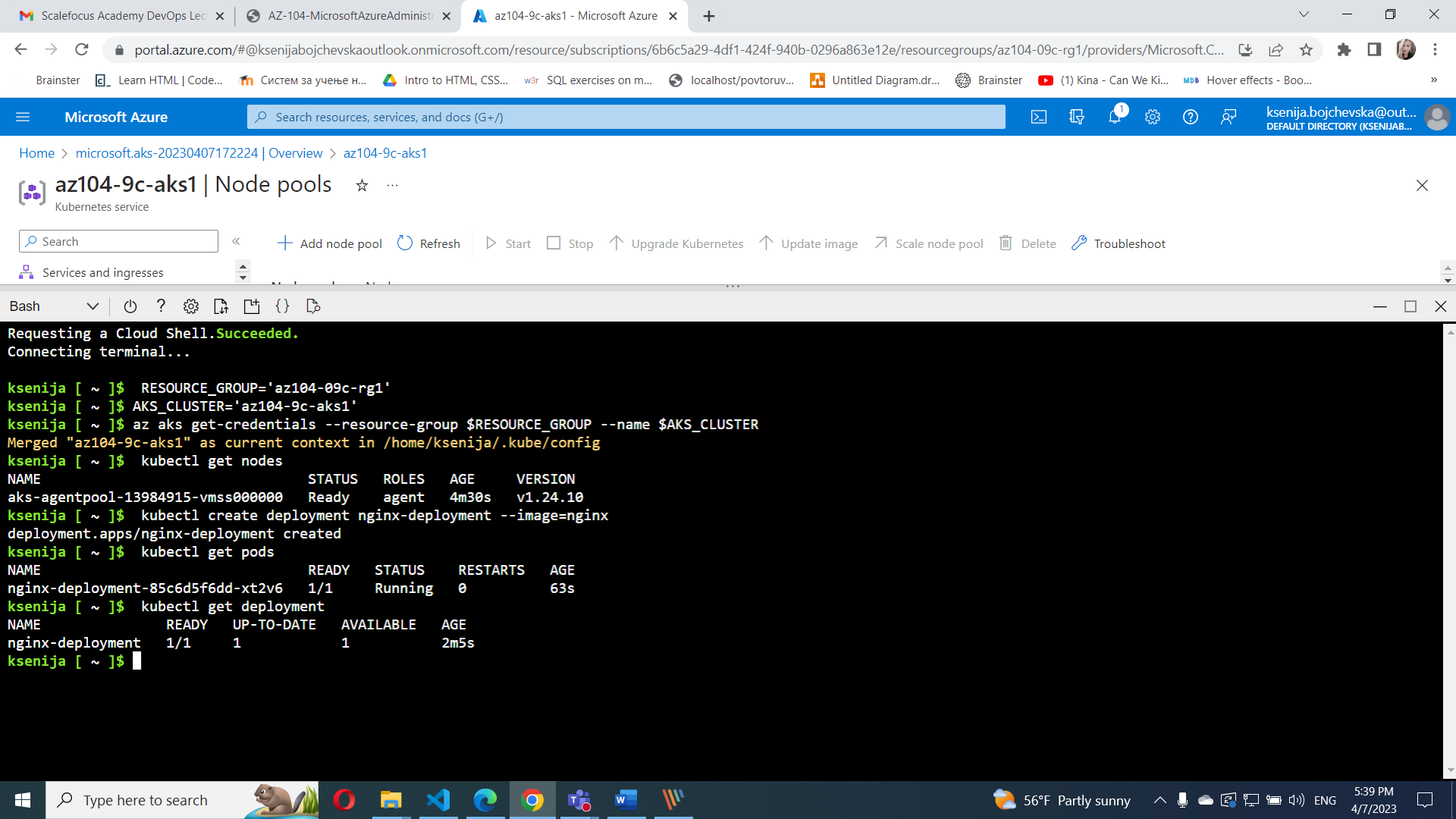
kubectl create deployment nginx-deployment --image=nginx



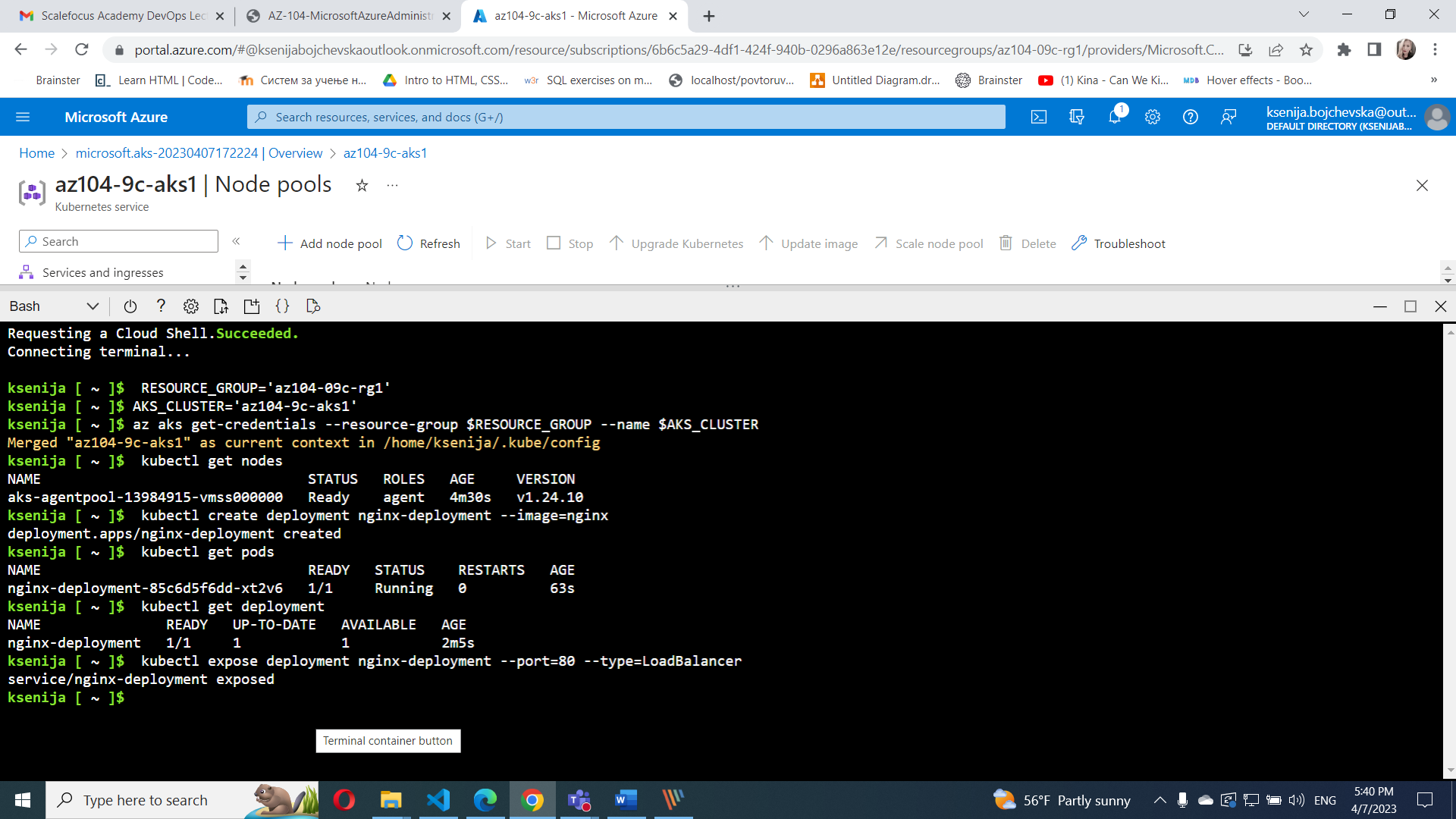
kubectl get pods



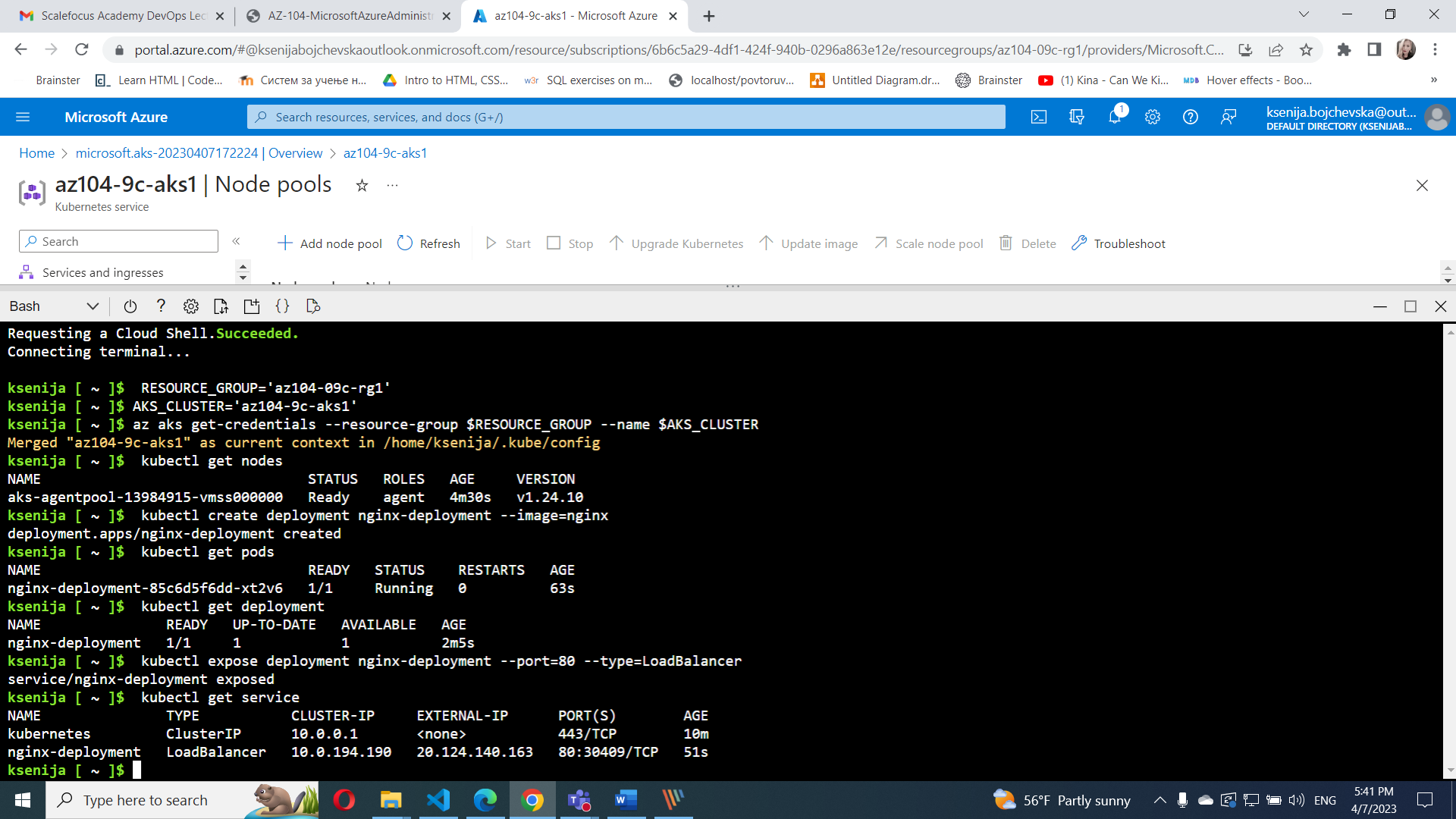
kubectl get deployment



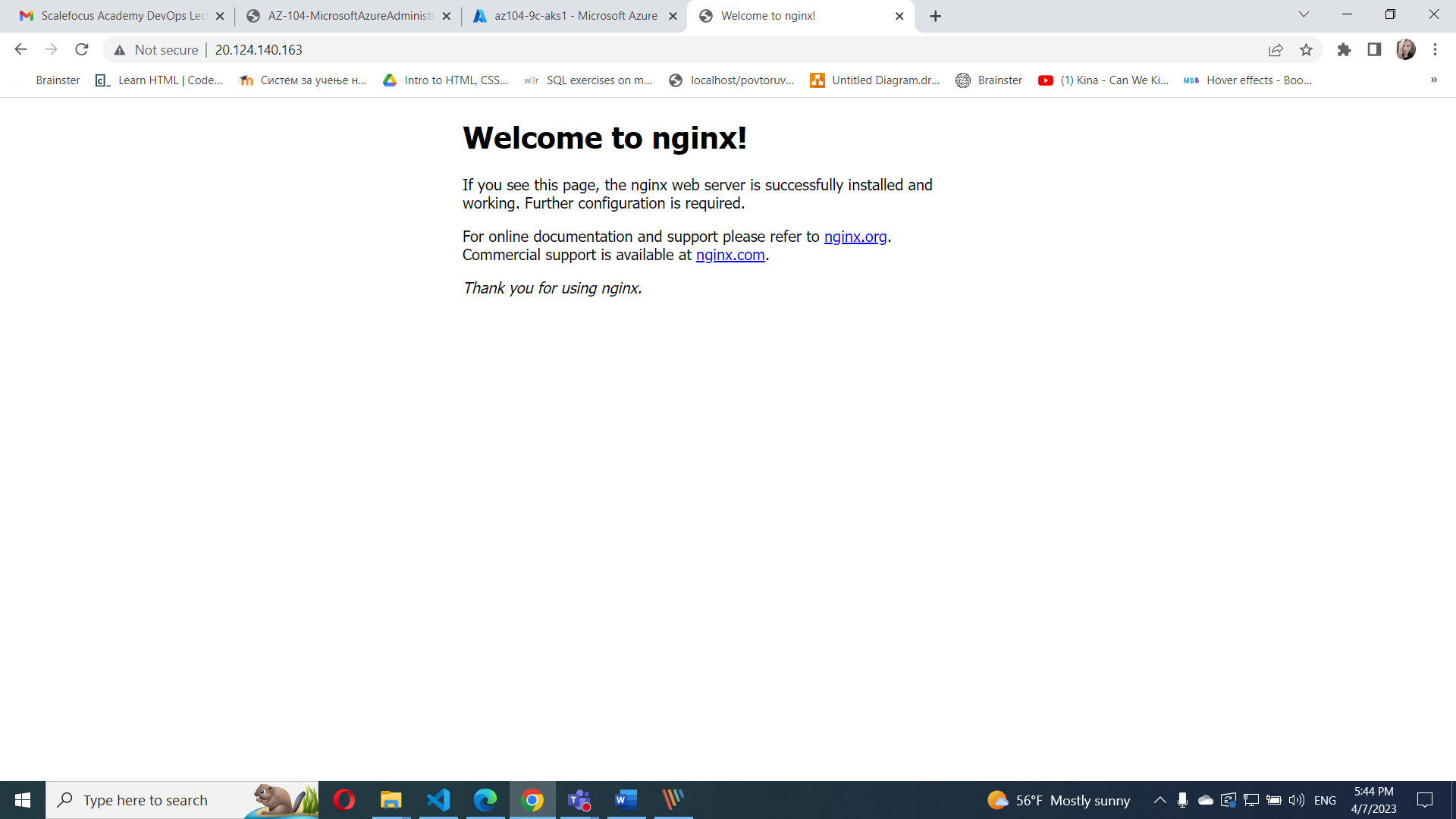
kubectl expose deployment nginx-deployment --port=80 --type=LoadBalancer



kubectl get service



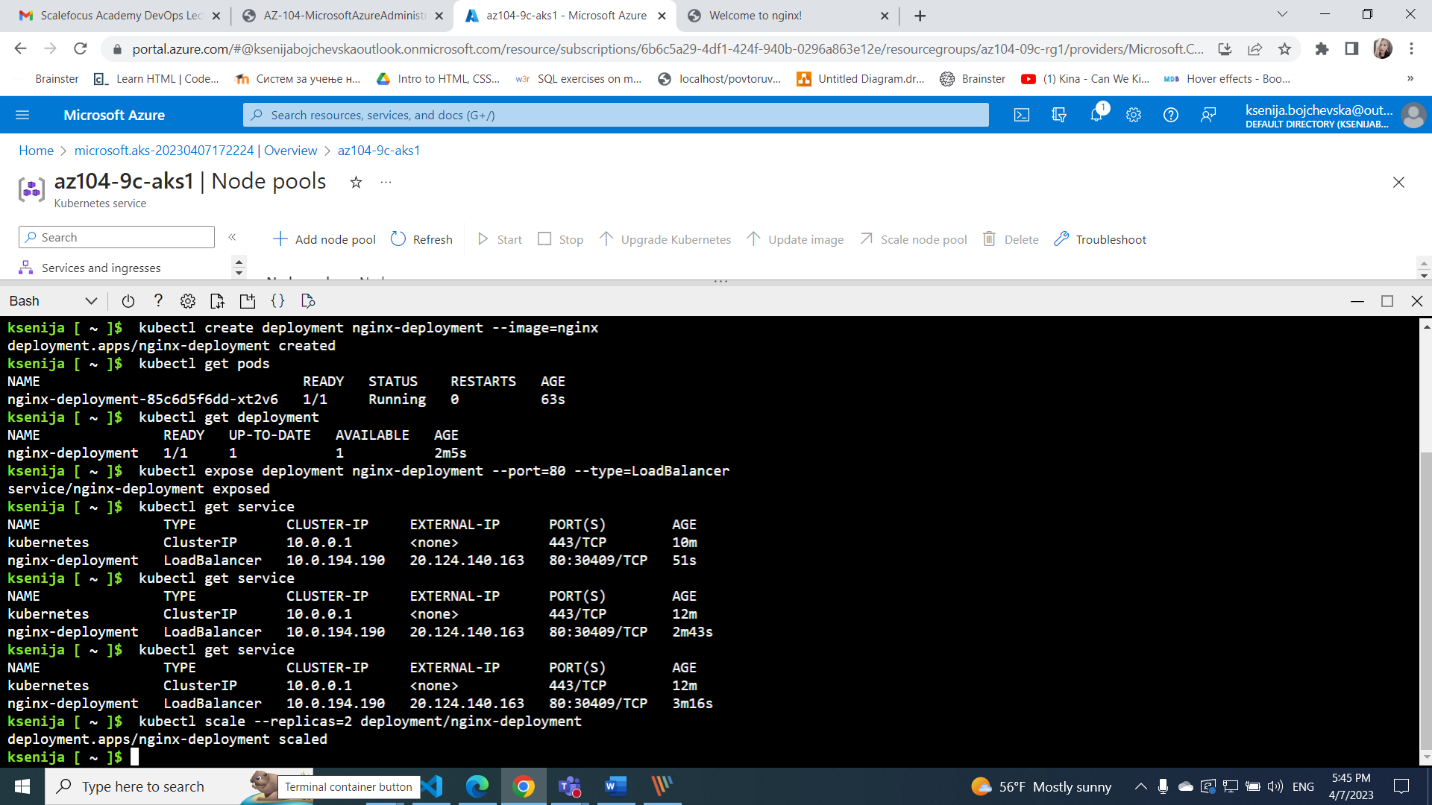
1. Re-run the command until the value in the **EXTERNAL-IP** column for the **nginx-deployment** entry changes from **<pending>** to a public IP address. Note the public IP address in the **EXTERNAL-IP** column for **nginx-deployment**.
2. Open a browser window and navigate to the IP address you obtained in the previous step. Verify that the browser page displays the **Welcome to nginx!** message.



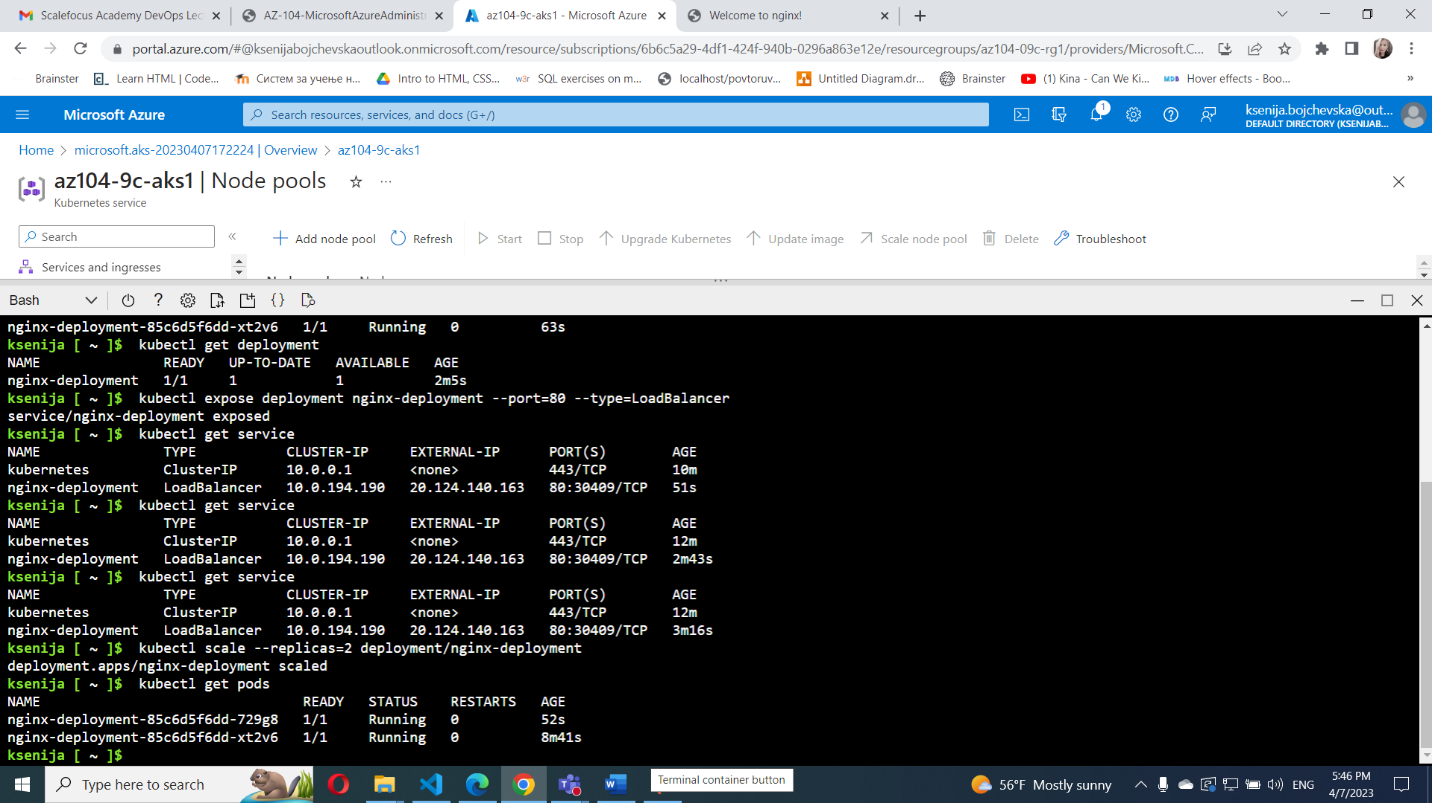
#### **Task 4: Scale containerized workloads in the Azure Kubernetes service cluster**

From the **Cloud Shell** pane, and run the following to scale the deployment by increasing of the number of pods to 2:

kubectl scale --replicas=2 deployment/nginx-deployment



kubectl get pods

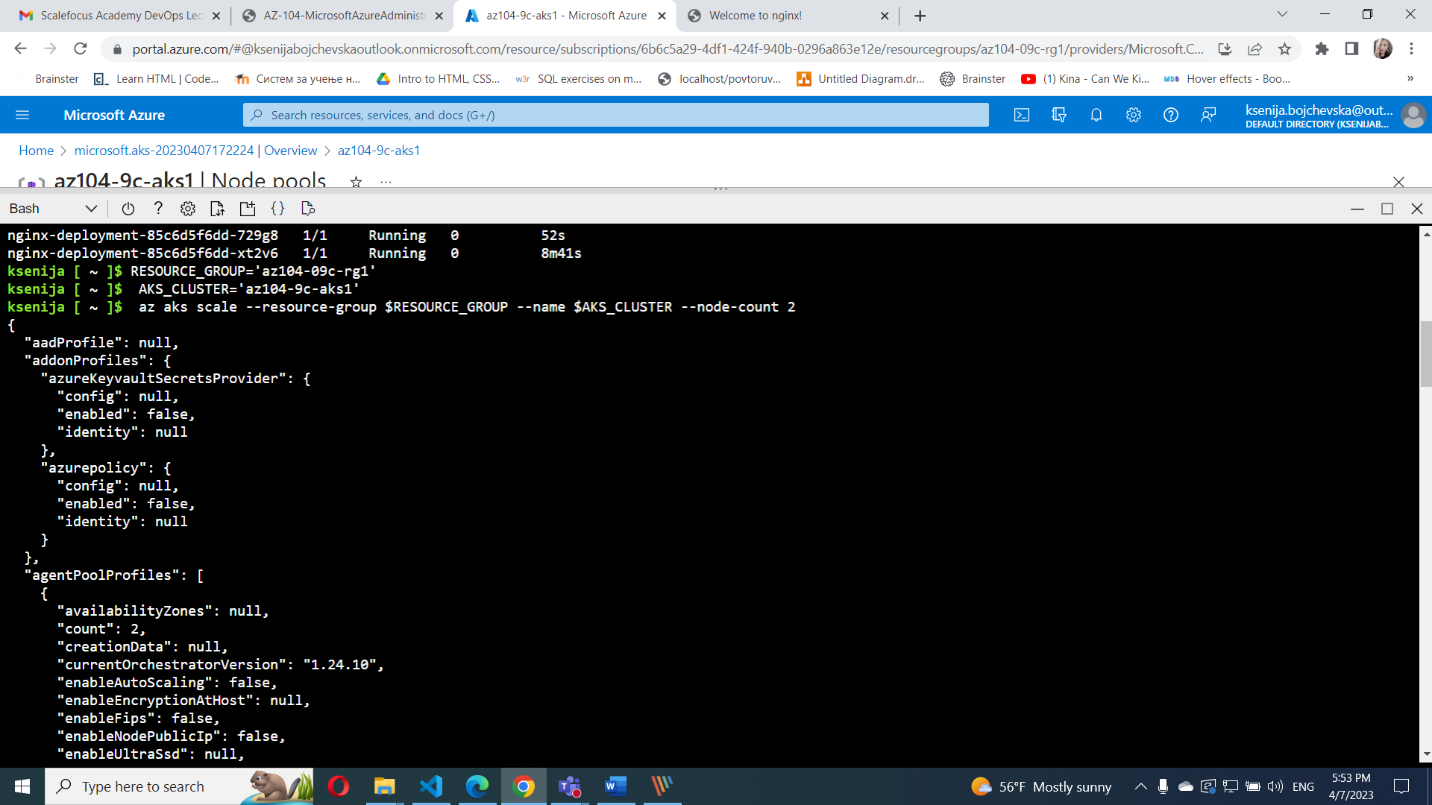


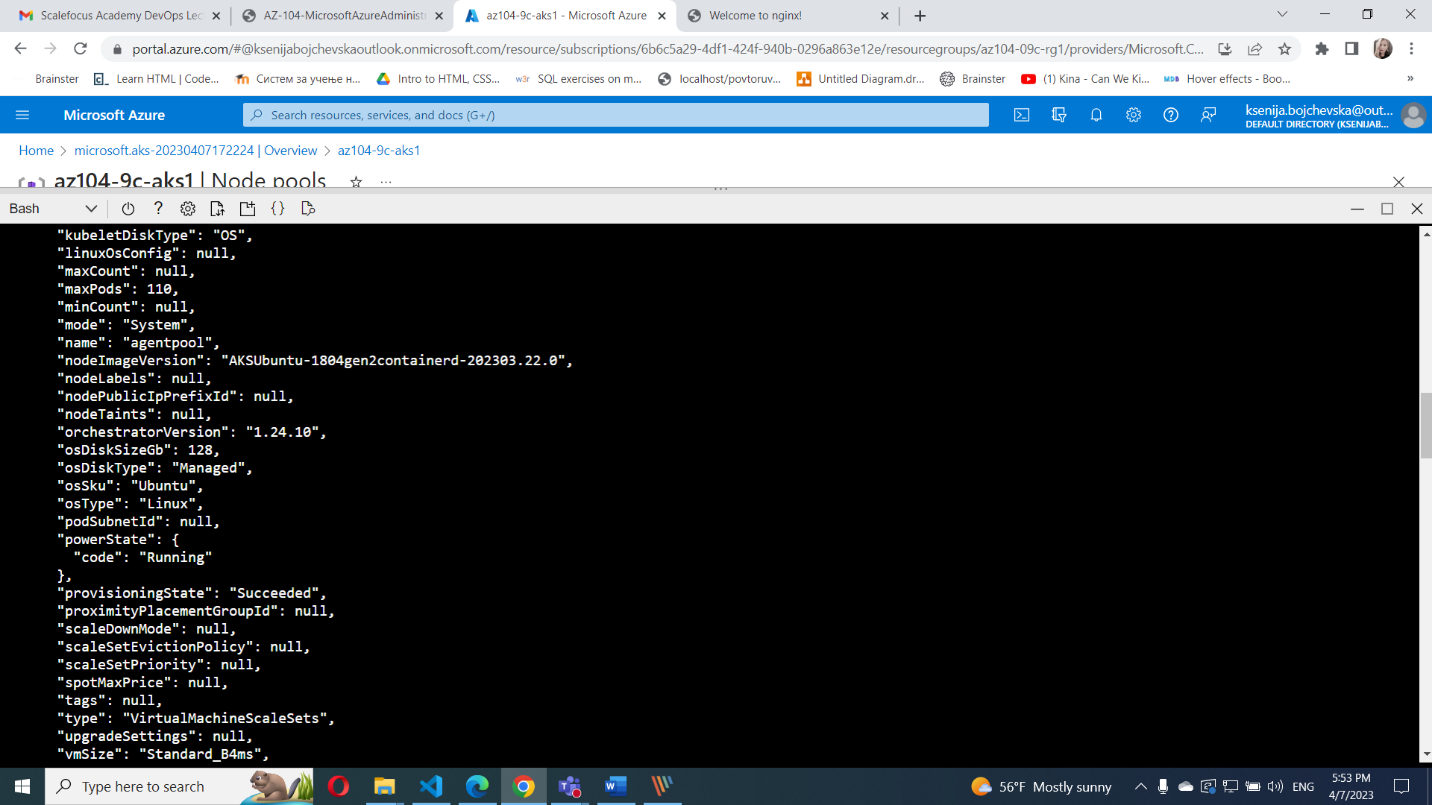
From the **Cloud Shell** pane, run the following to scale out the cluster by increasing the number of nodes to 2:

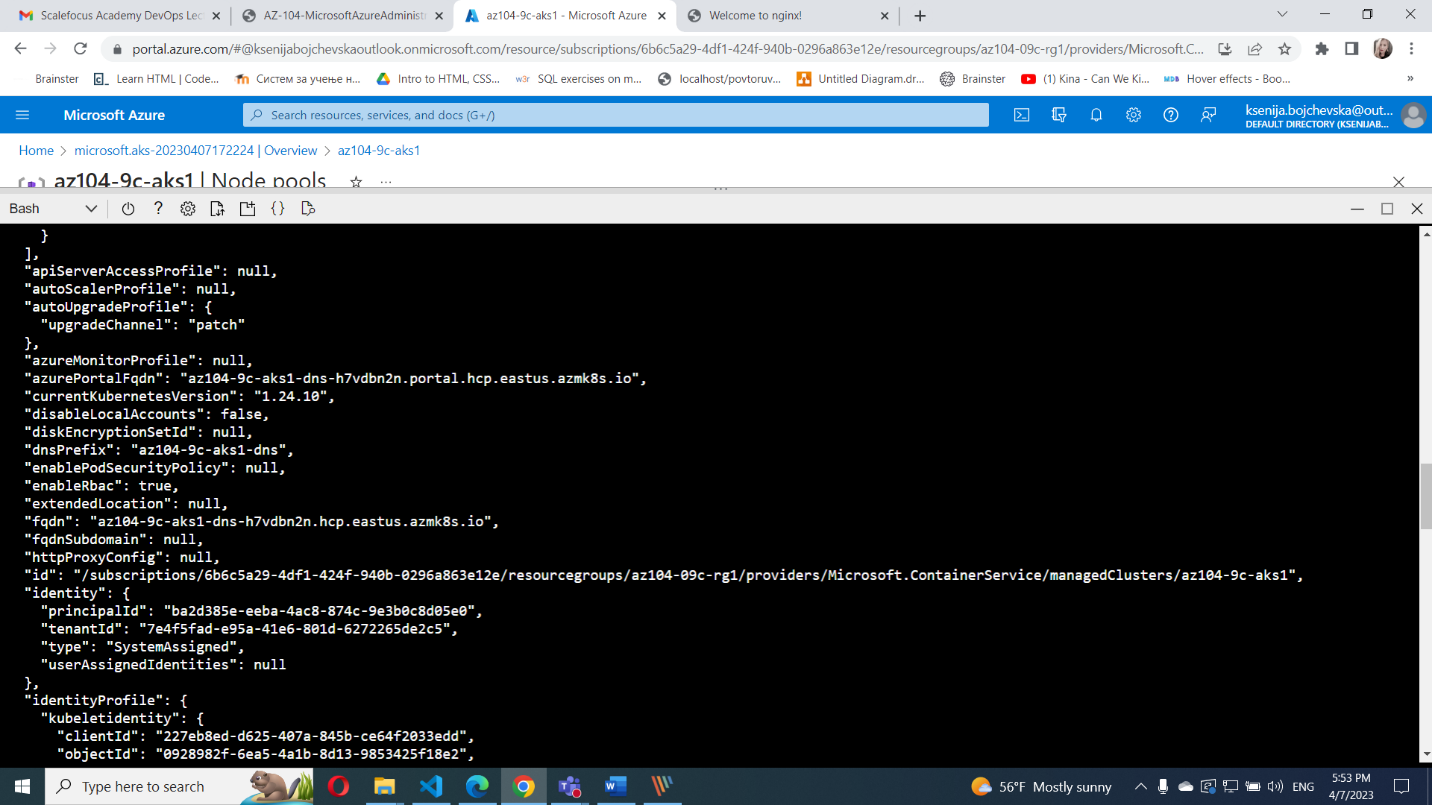
RESOURCE\_GROUP='az104-09c-rg1'

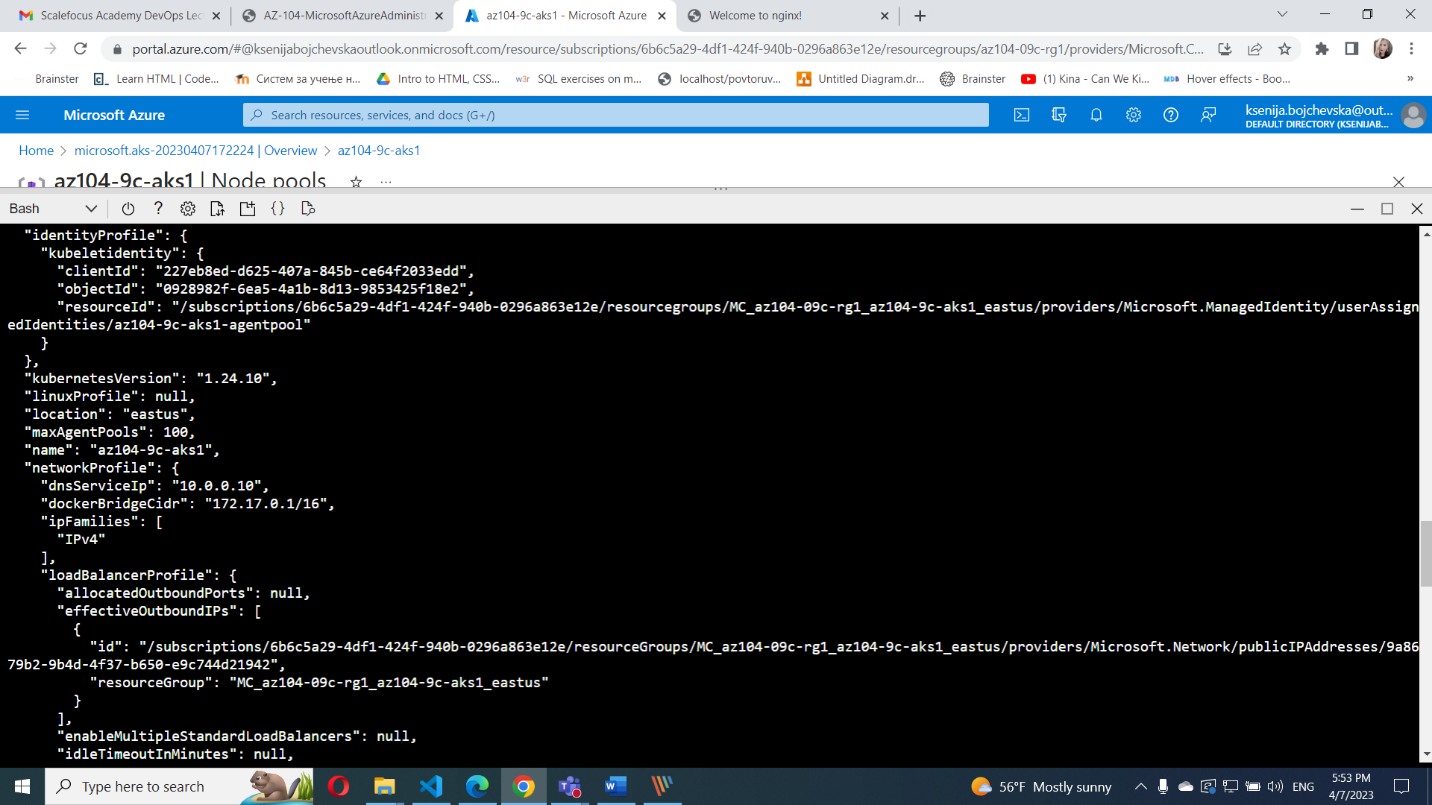
AKS\_CLUSTER='az104-9c-aks1'

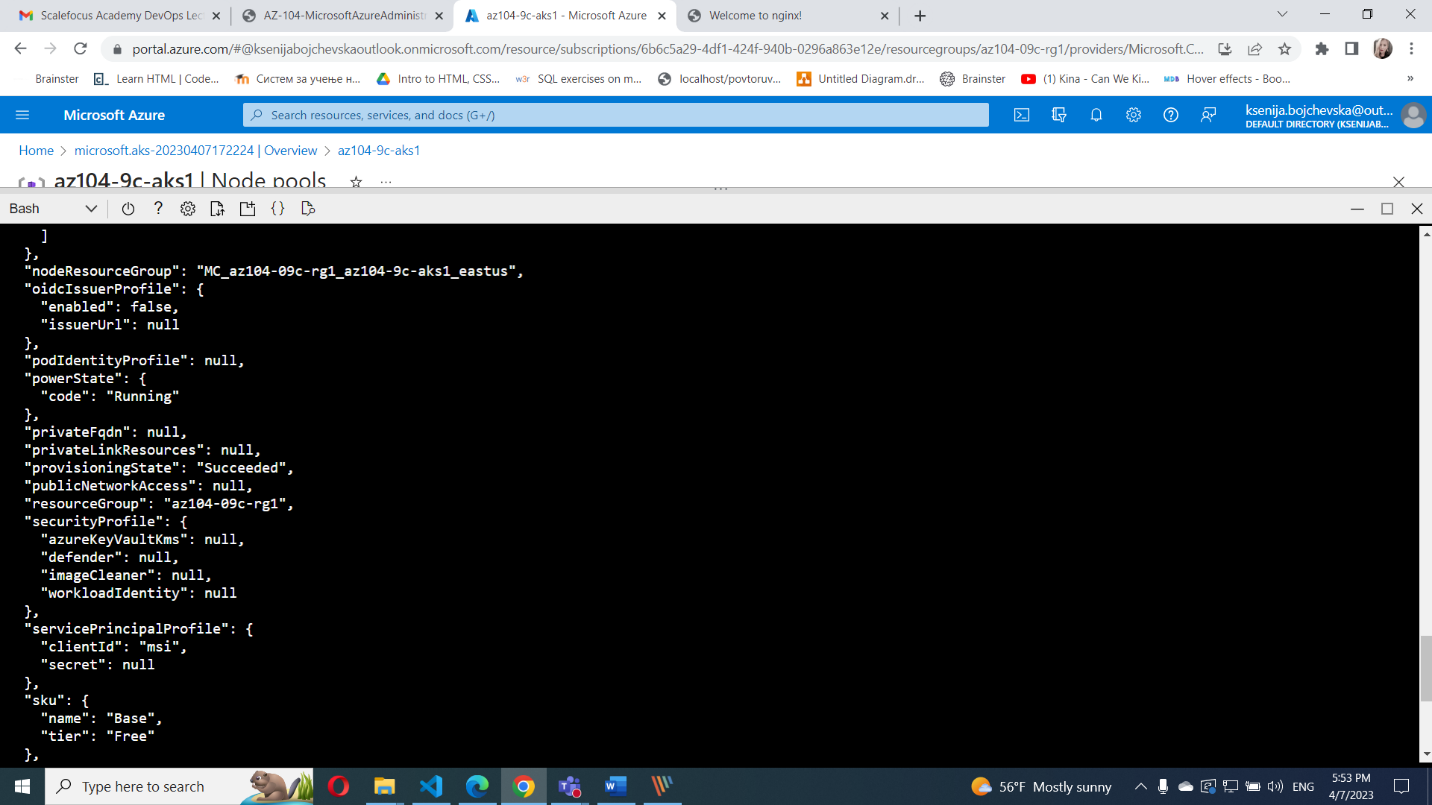
az aks scale --resource-group $RESOURCE\_GROUP --name $AKS\_CLUSTER --node-count 2

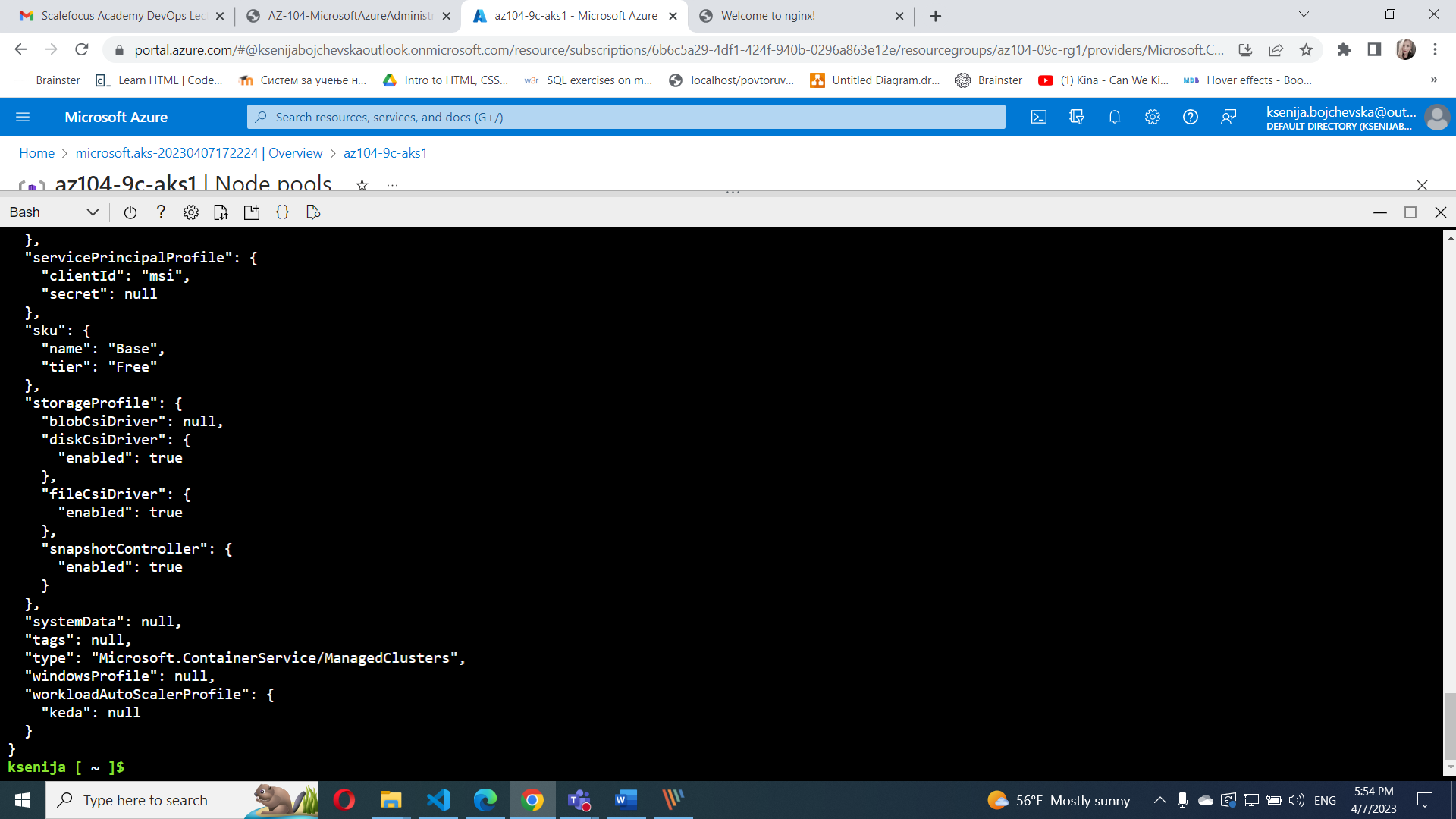






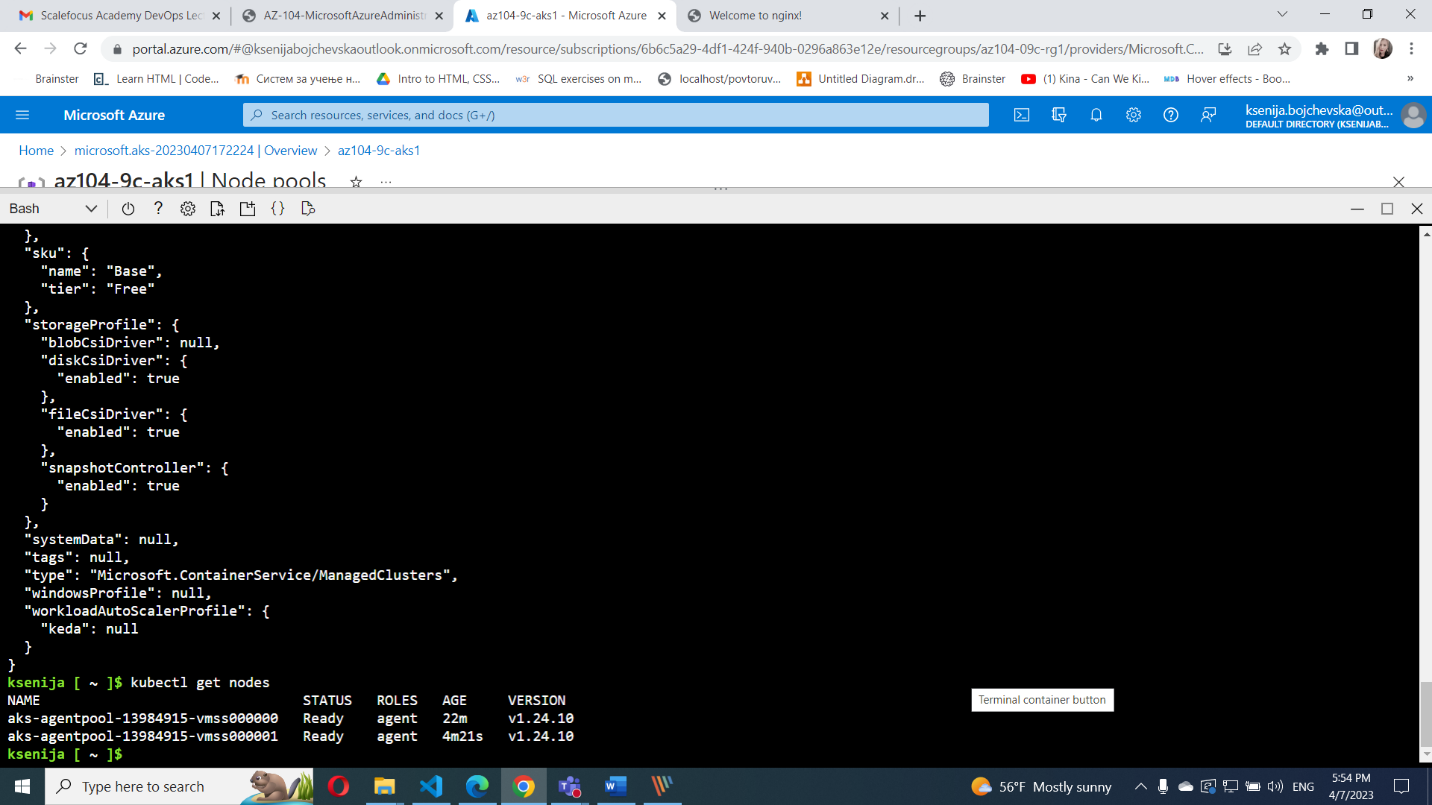




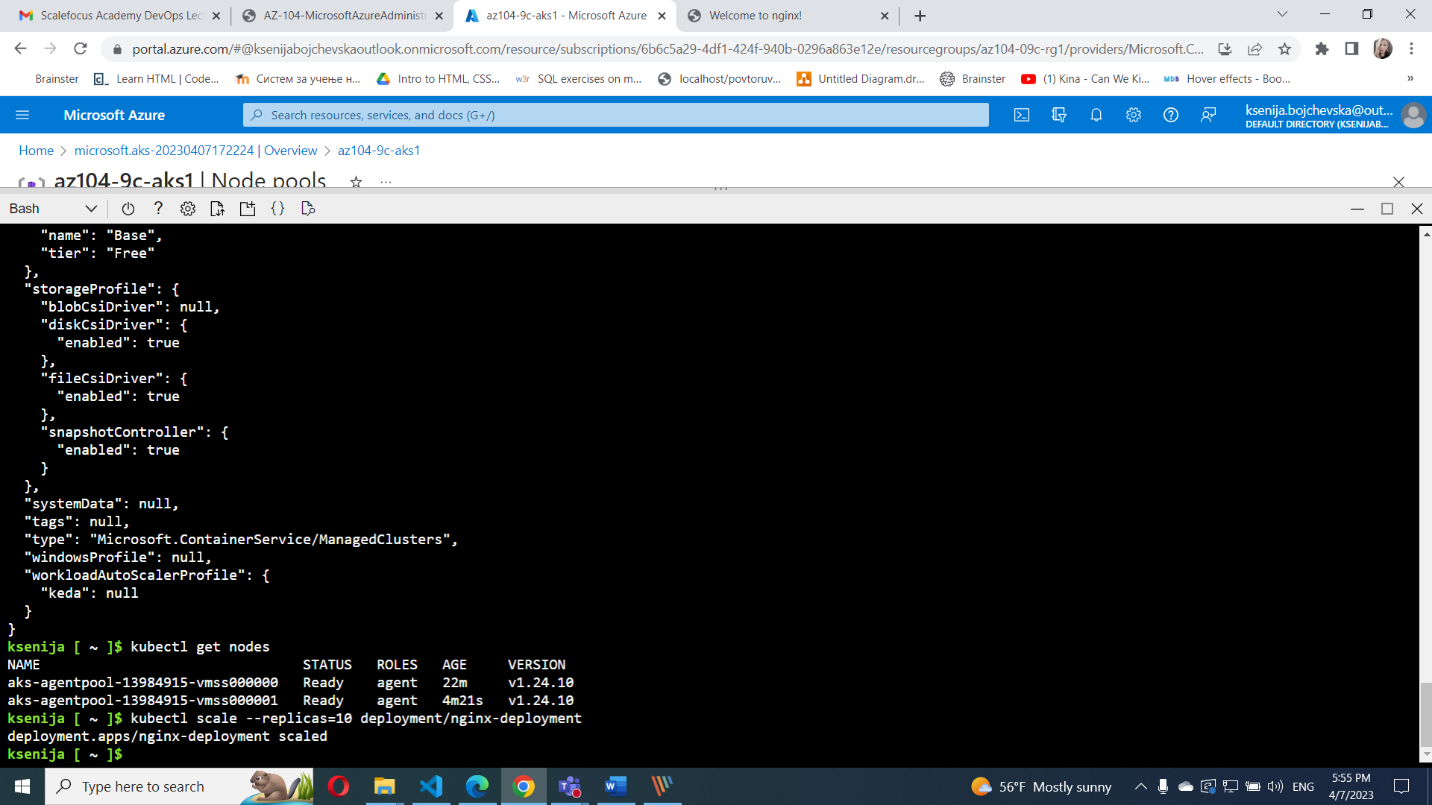


From the **Cloud Shell** pane, run the following to verify the outcome of scaling the cluster:

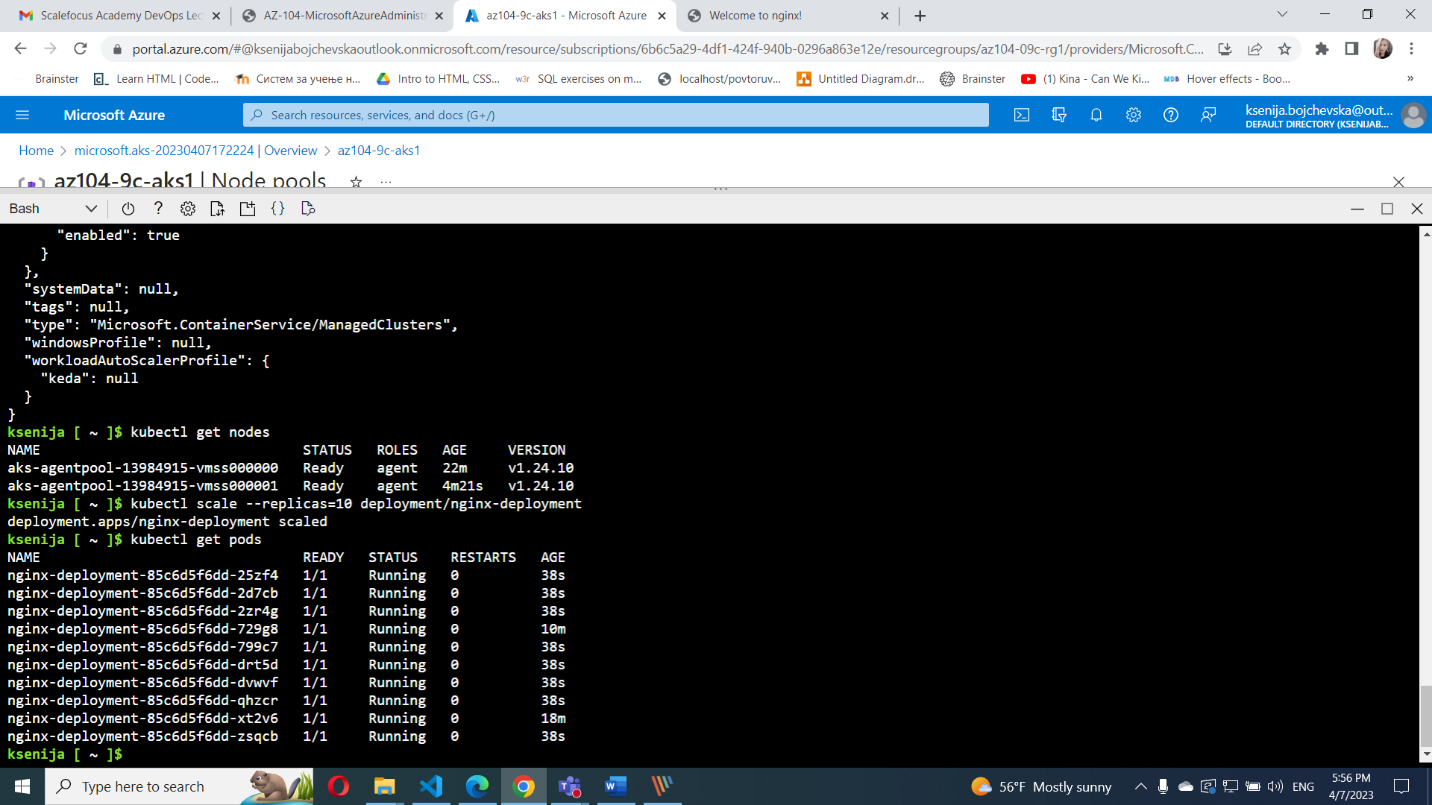
kubectl get nodes



kubectl scale --replicas=10 deployment/nginx-deployment



kubectl get pods



kubectl get pod -o=custom-columns=NODE:.spec.nodeName,POD:.metadata.name

