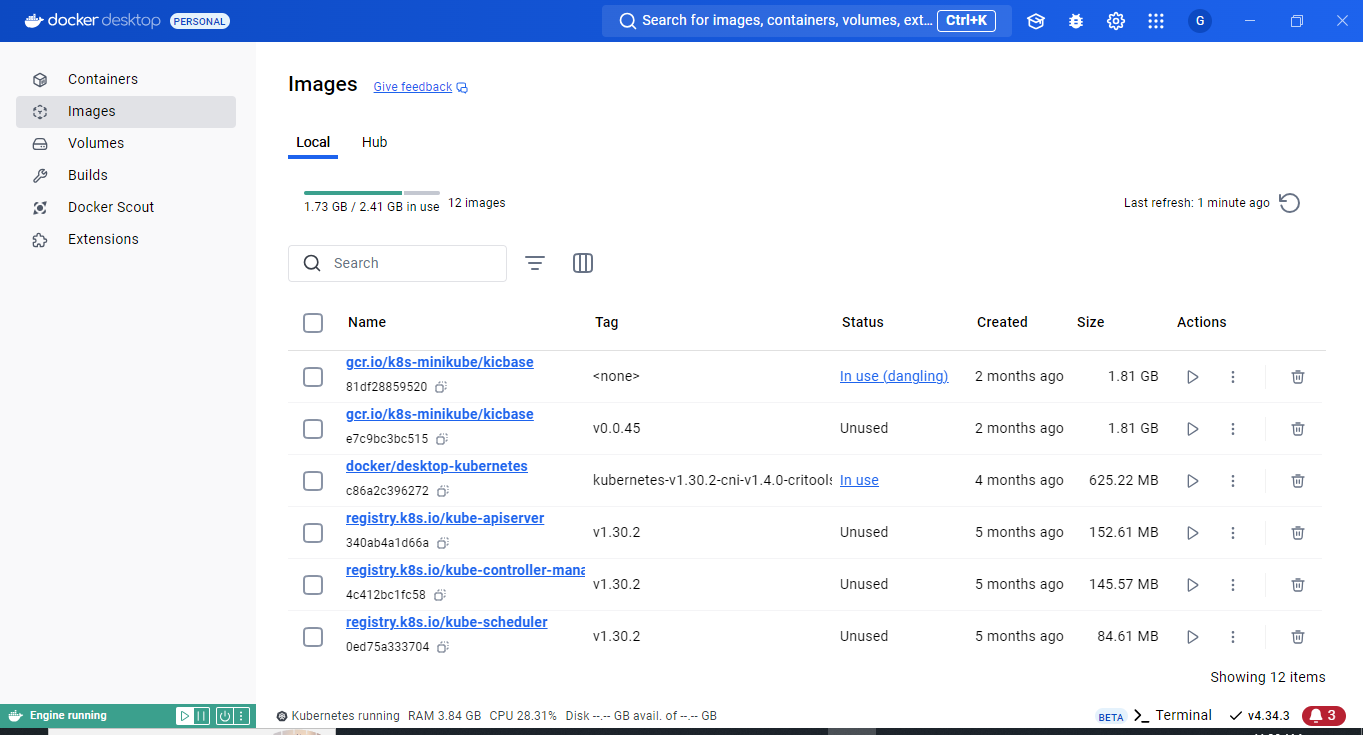
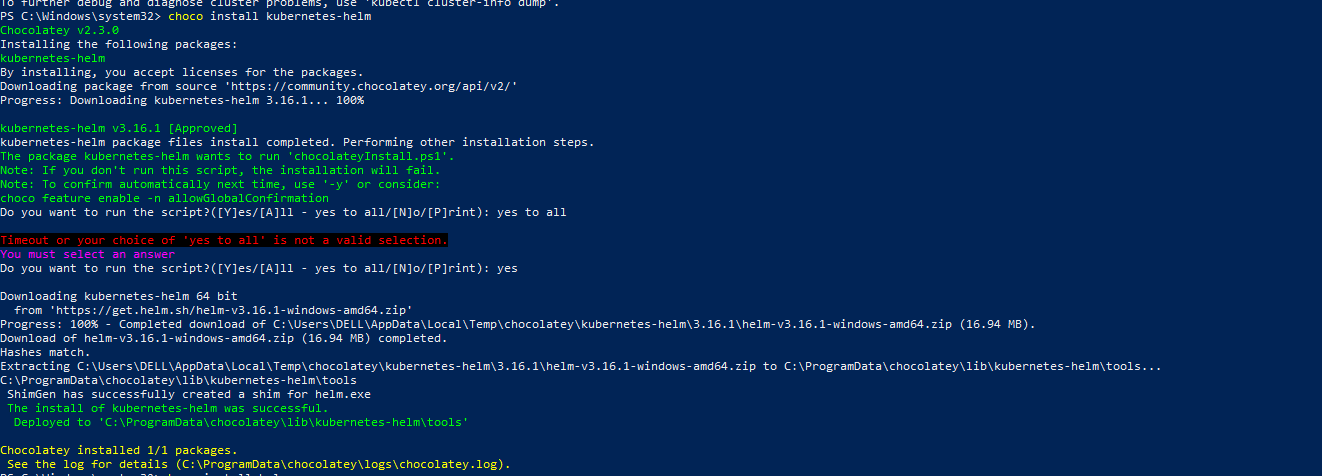
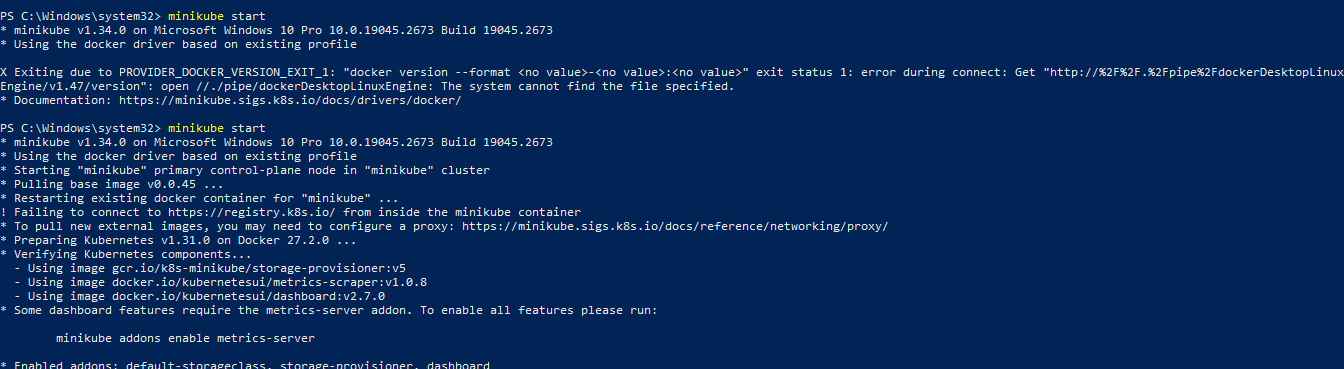
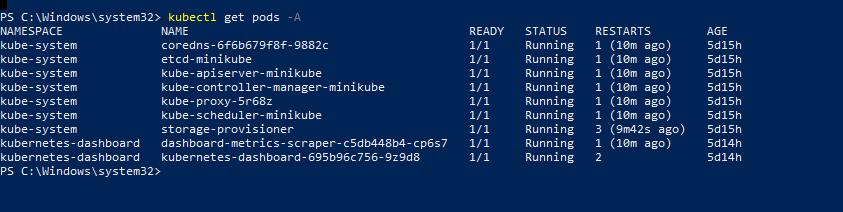
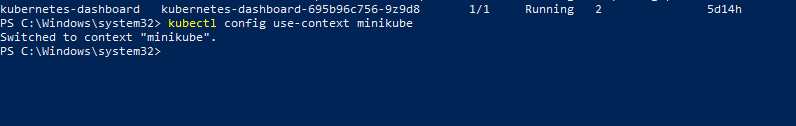
Kubernetess helm

  
first step we need to start docker engine and install helm in our machine by open windows powershell with admin   
step 1: run this command.  
choco install kubernetes-helm -y  
example  


Step 2: start minikube by using this command minikube start  
all before that you need to start your docker first and this is the most important thing ever. Now lets start minikube.  
  
check all the default pods are running by using this command  
kubectl get pods –A  
  


If you want to install kube into a single one by using this command

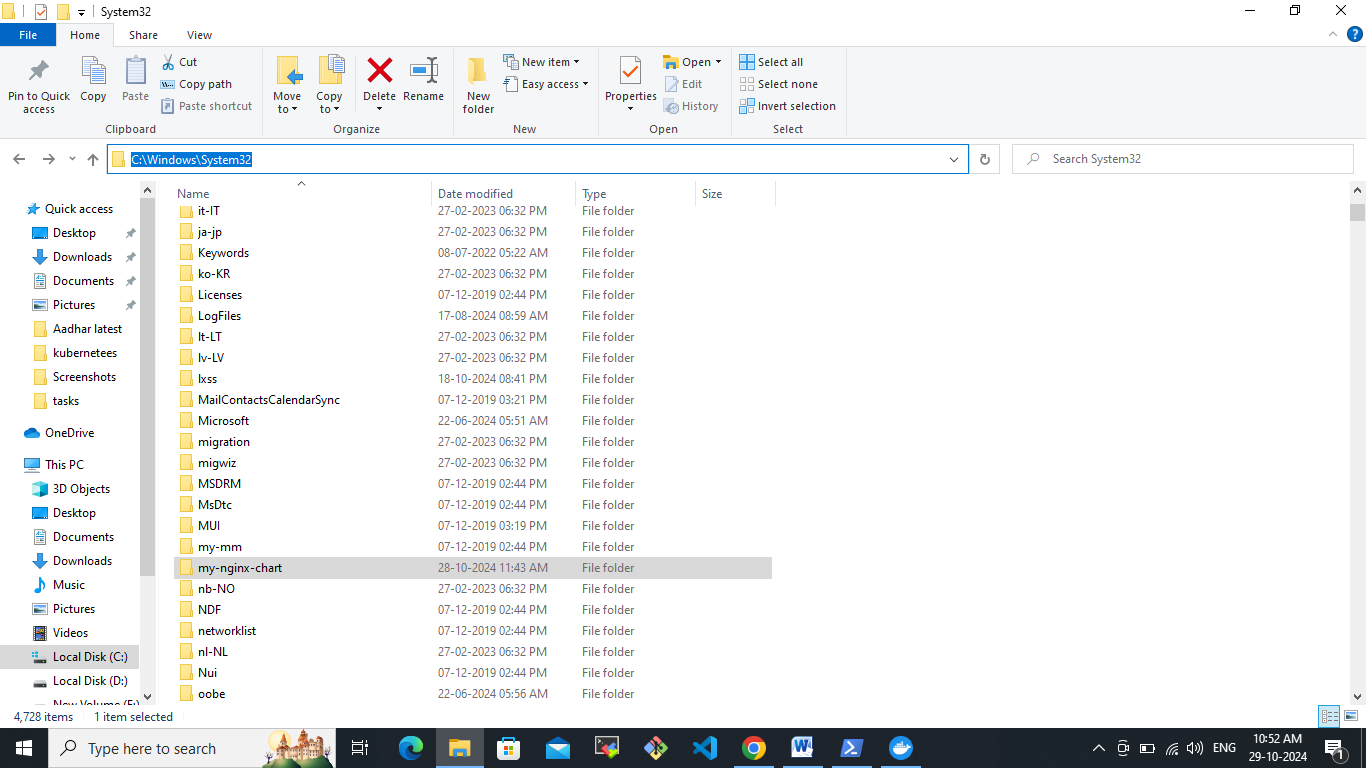
Kubectl config use-context minikube

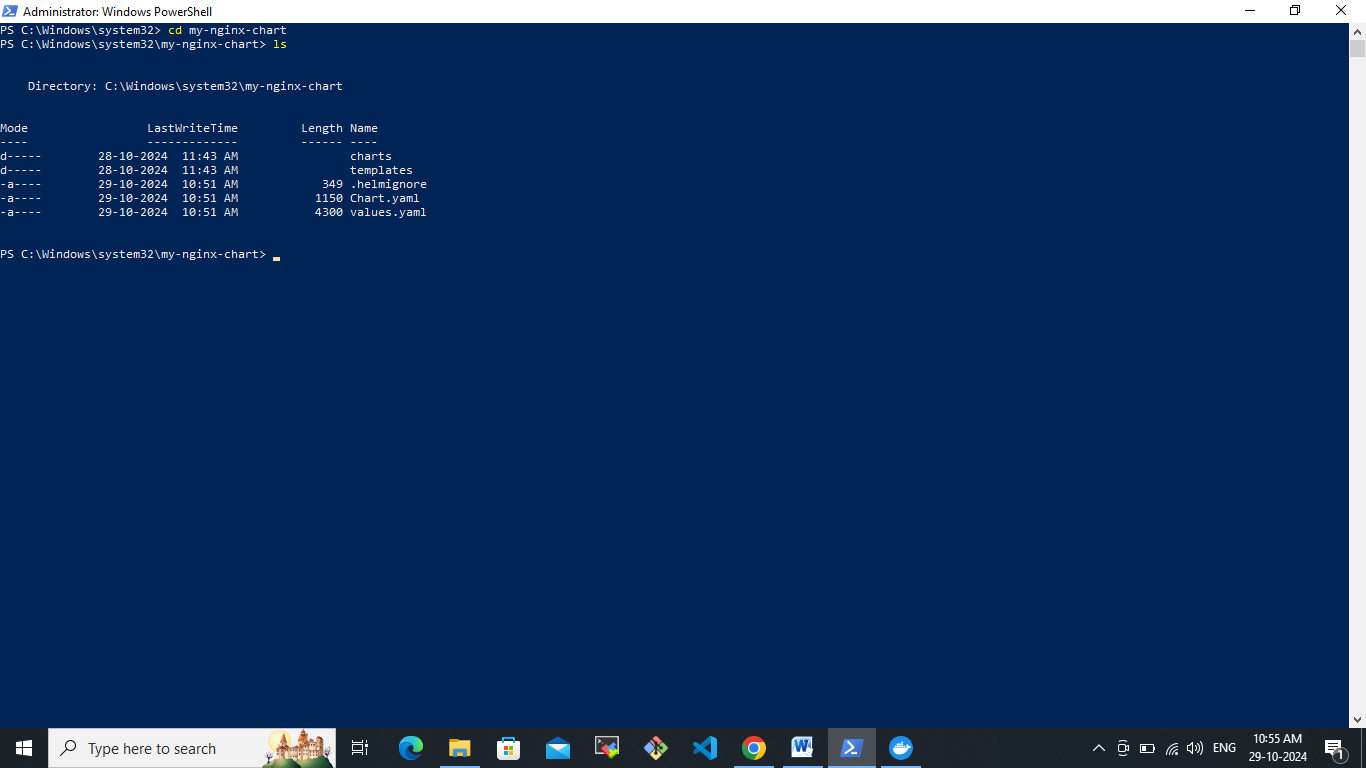


To create nginx by using helm chart by using this command

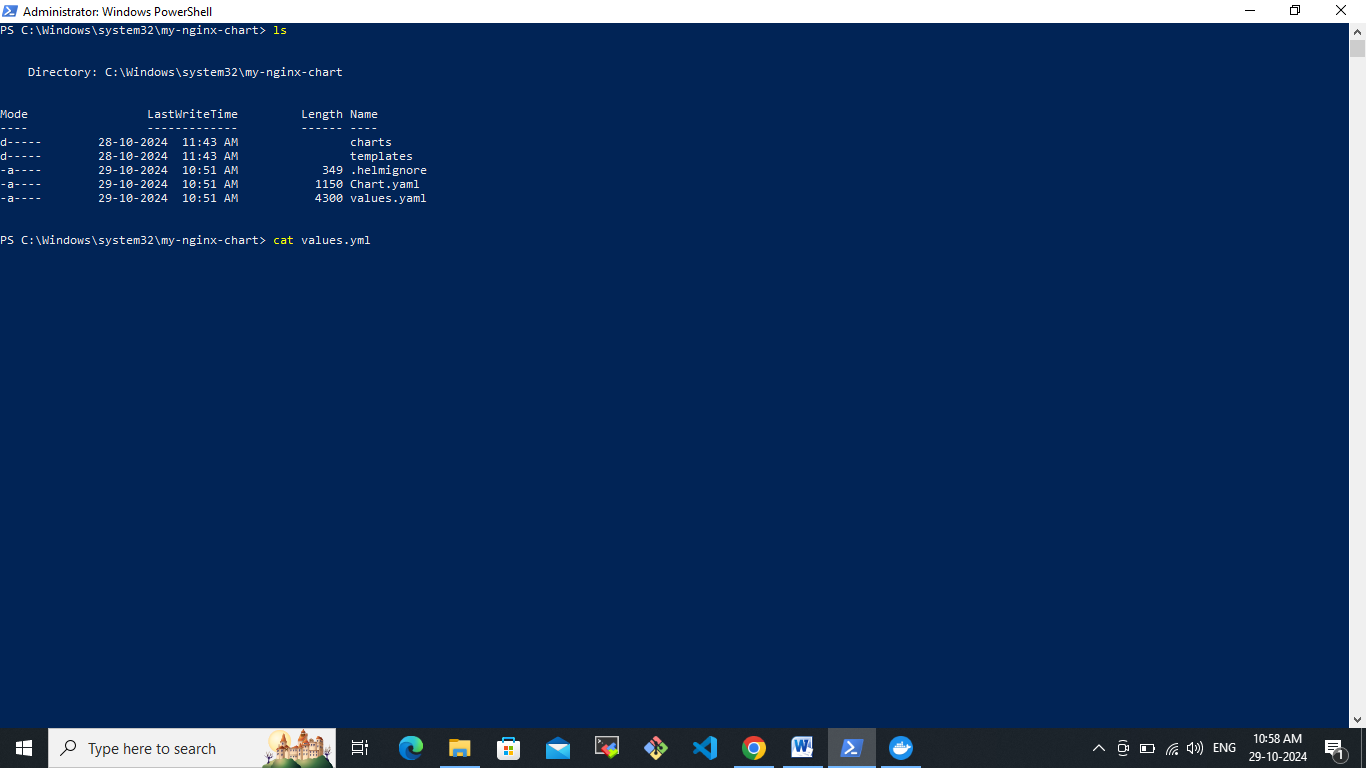
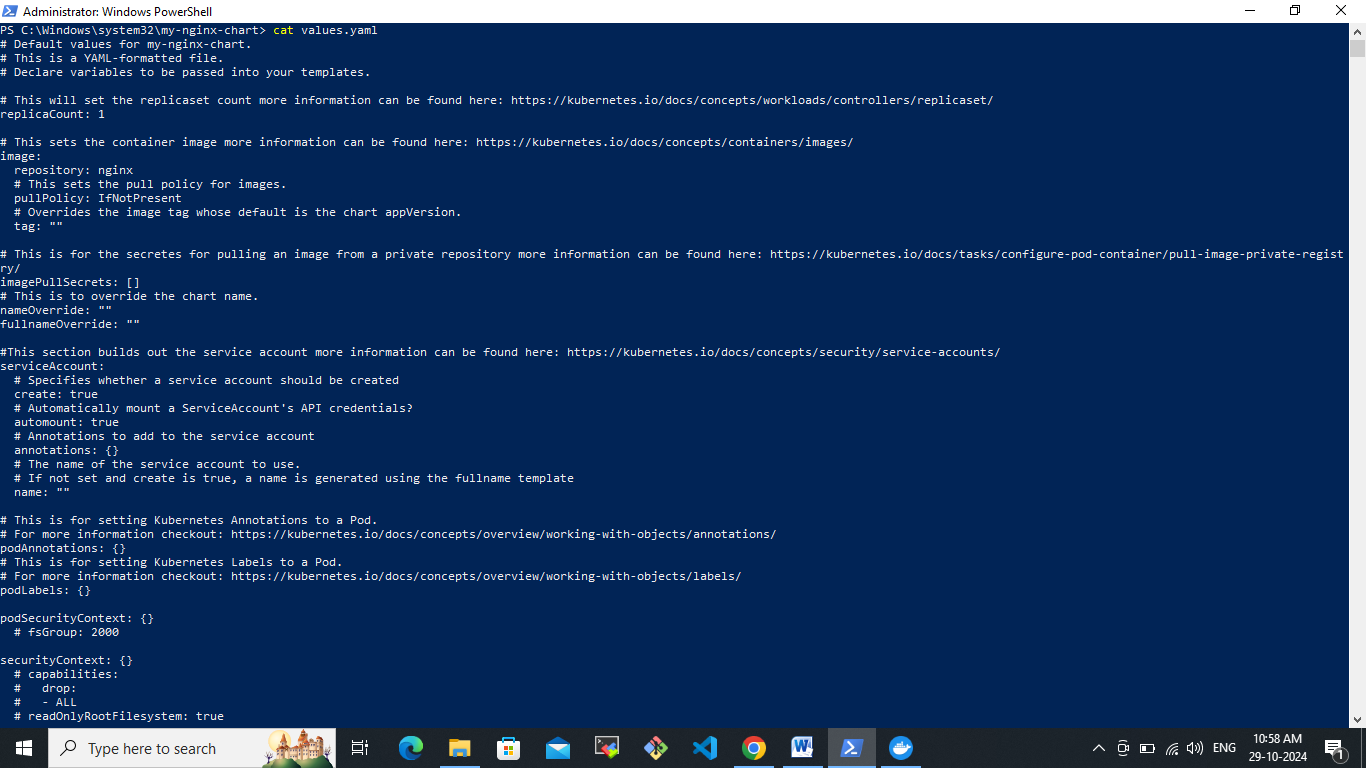


You can see by using ls on your powershell or you can see the directory

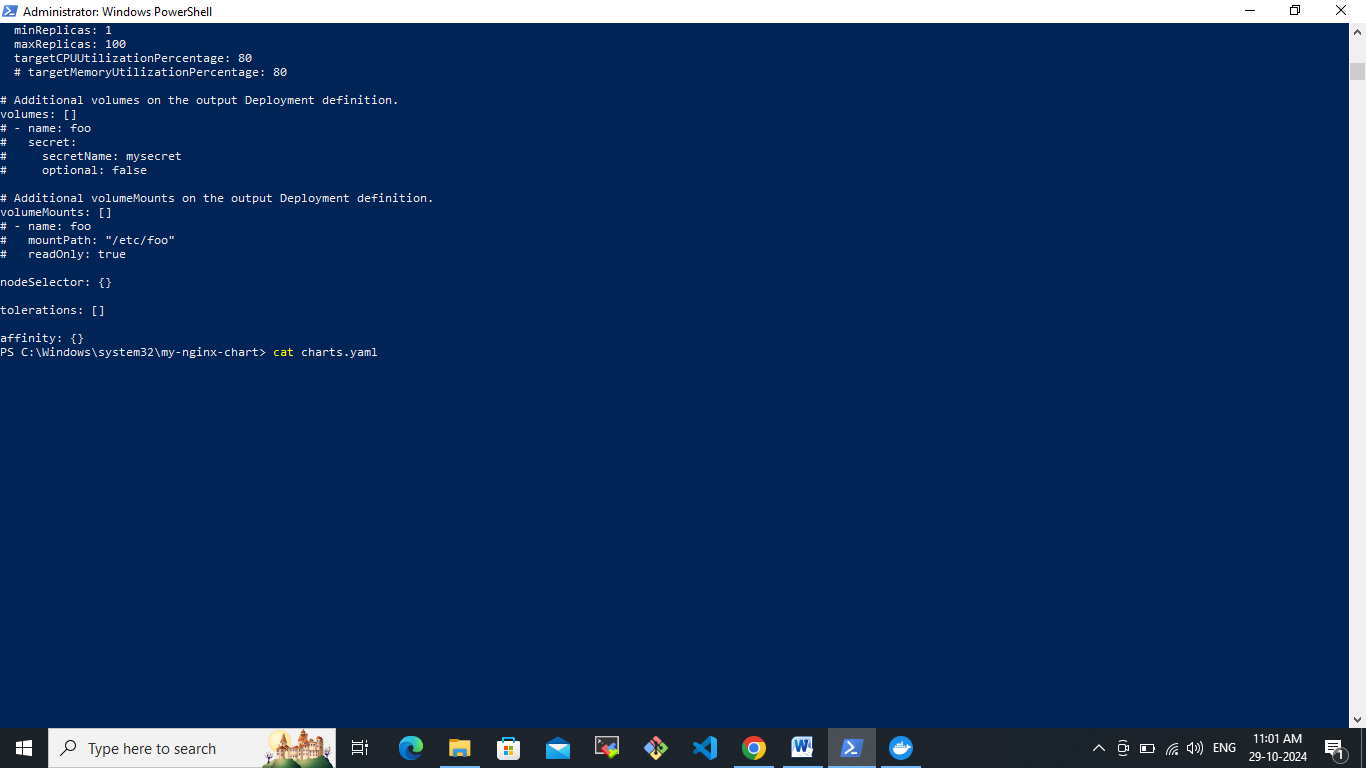
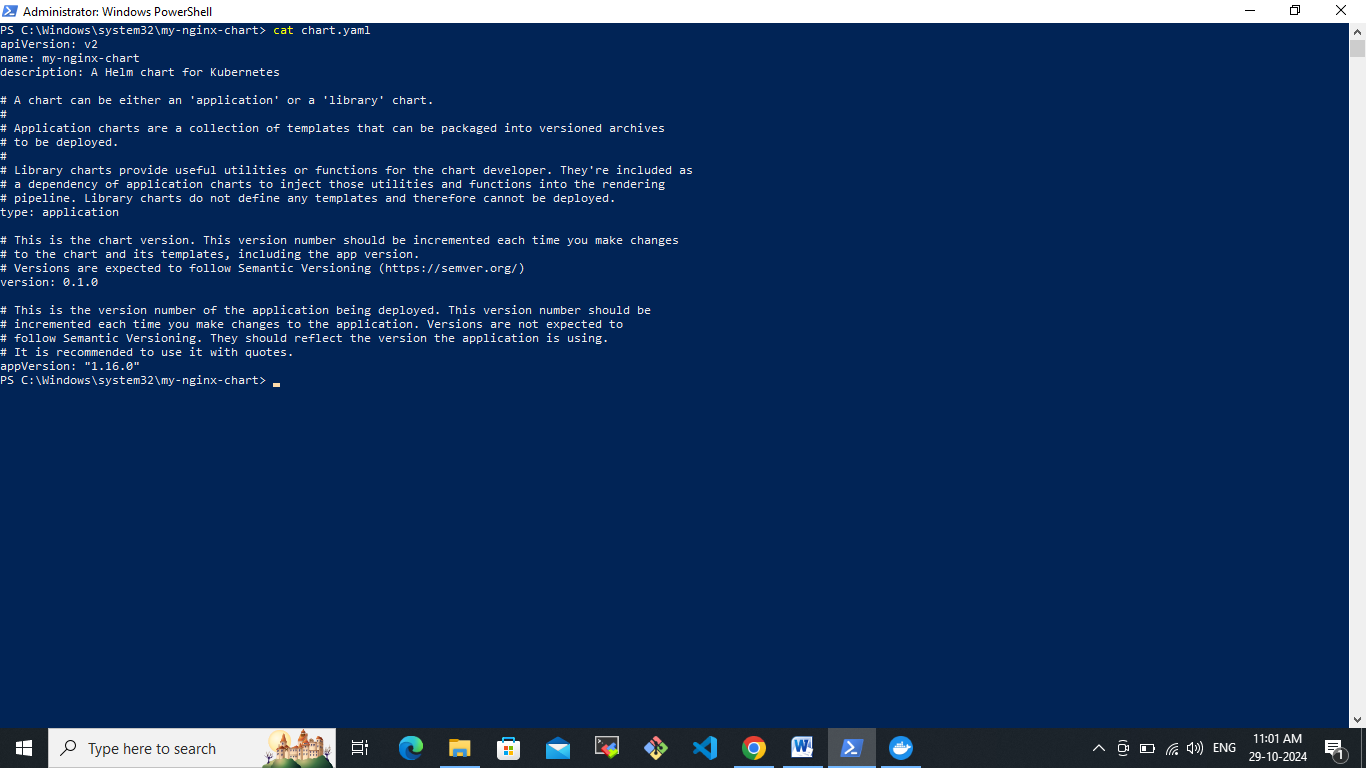




Let’s open values.yaml file **already they have defined their automatic structures. That is there. You can uncomment it or use it right? So it's already there. We will create new if you want, you can also use the existing configuration and walk around this right.**

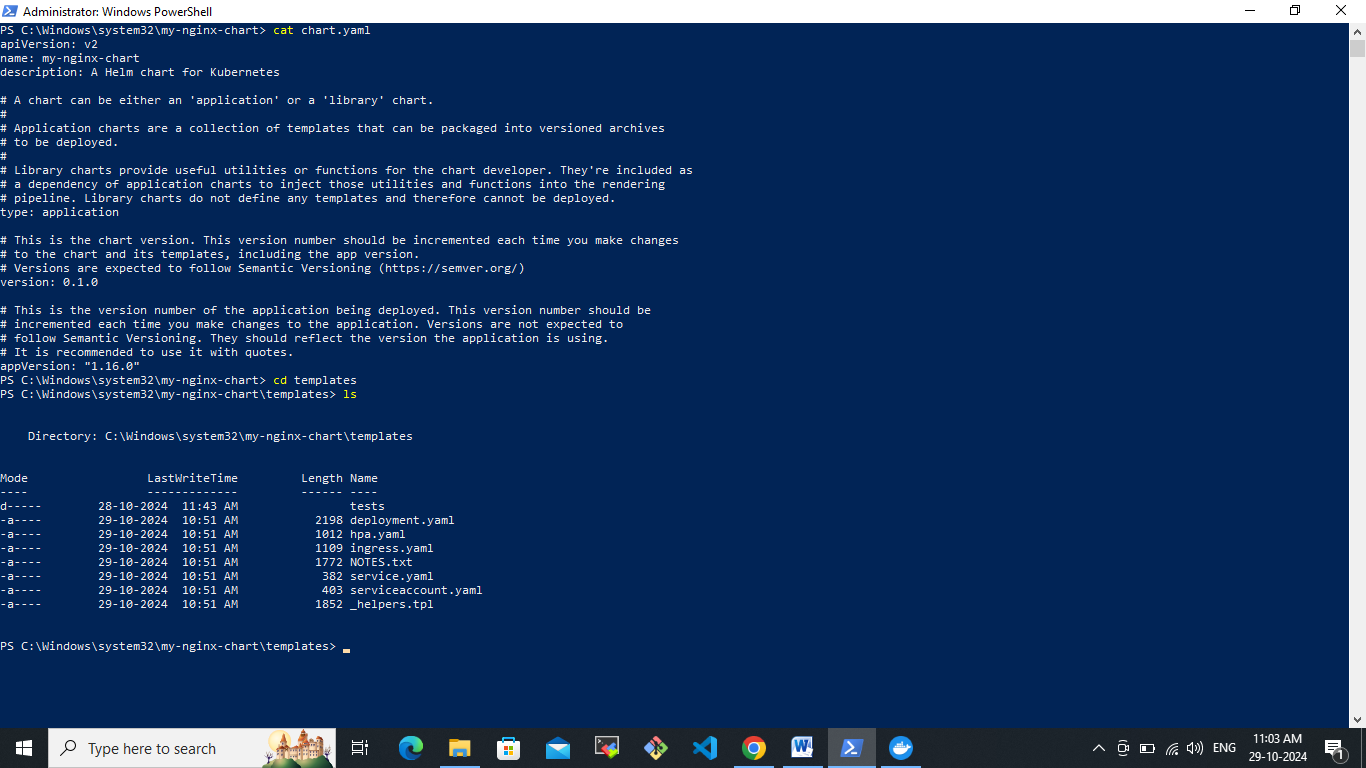
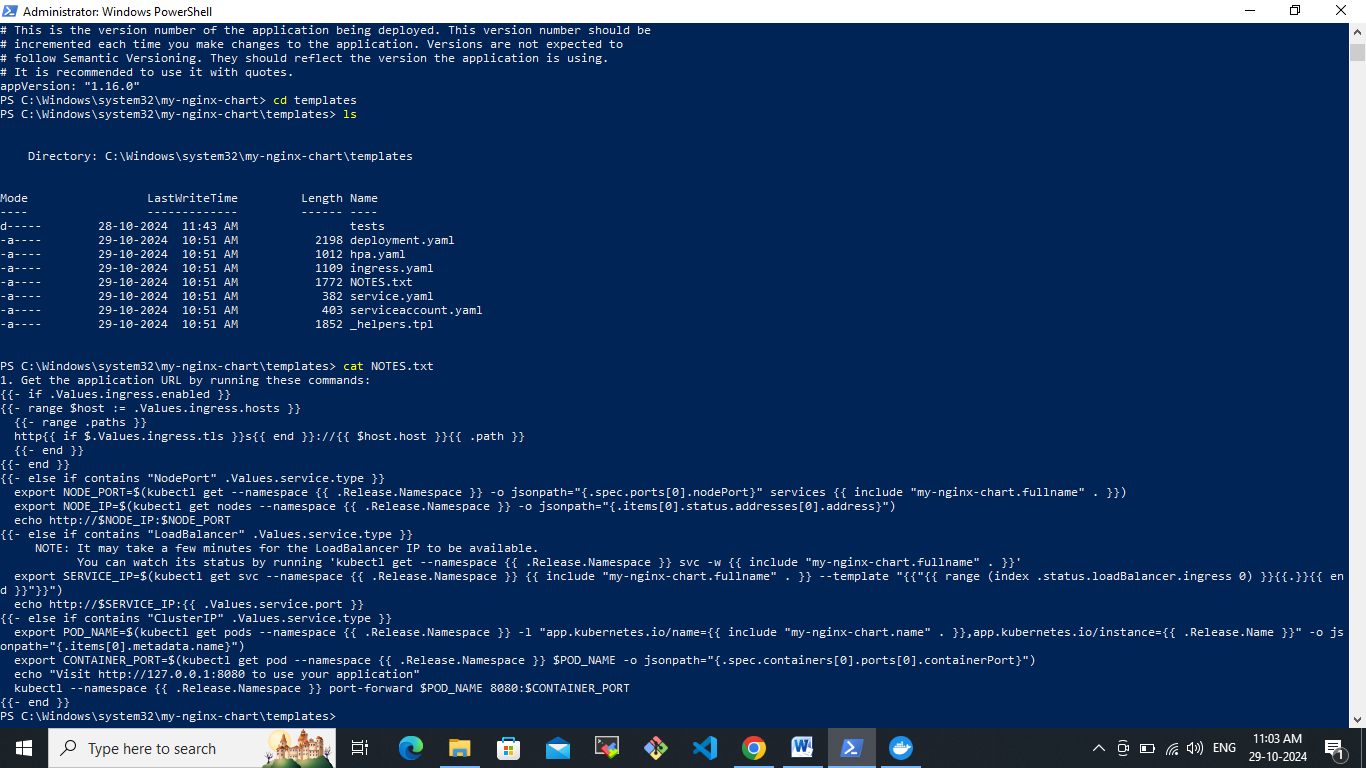
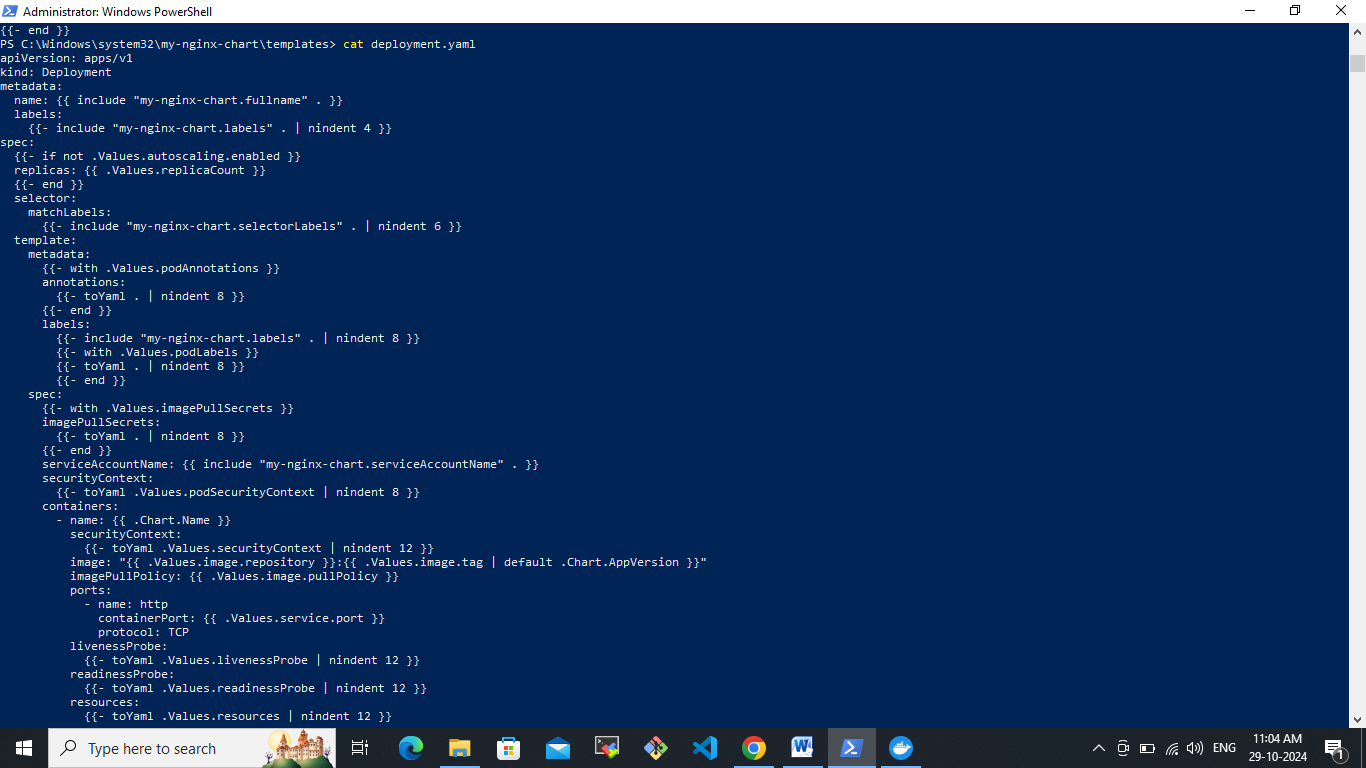
  


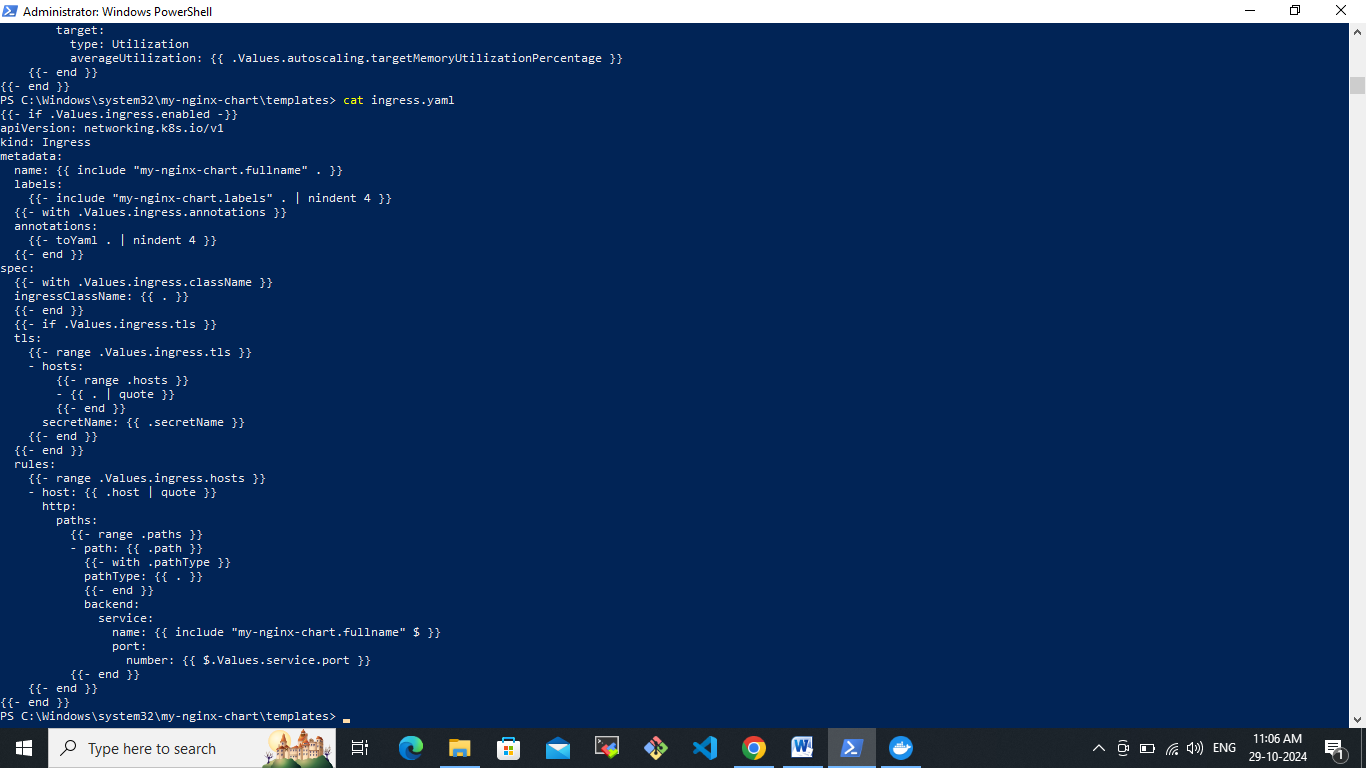
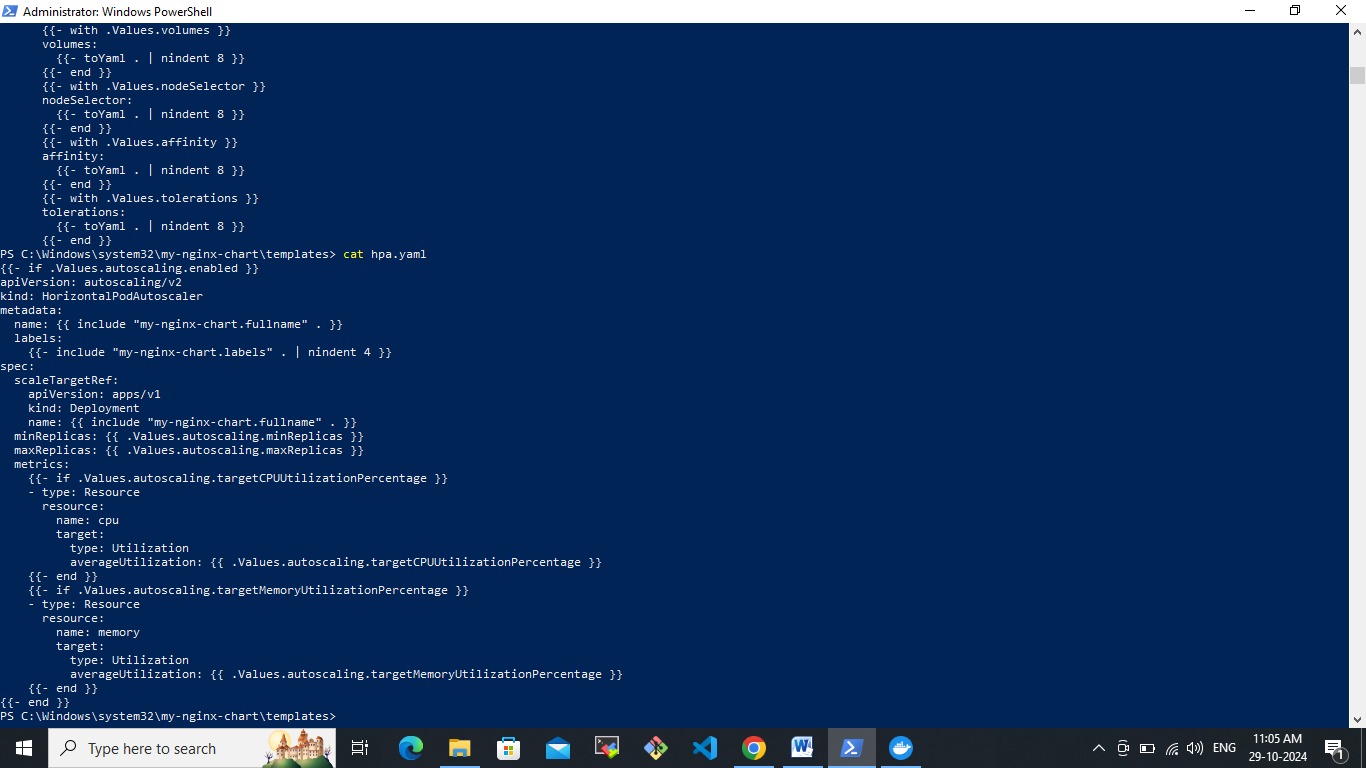
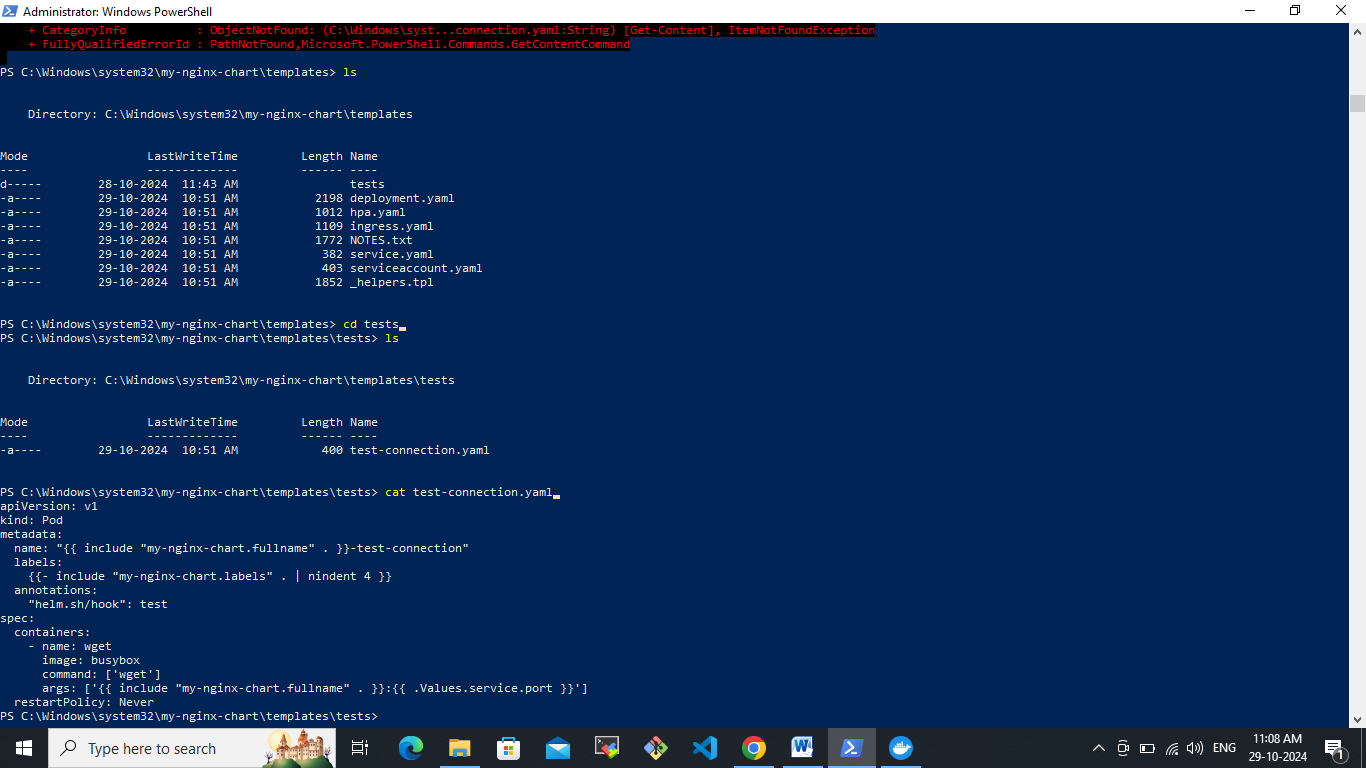
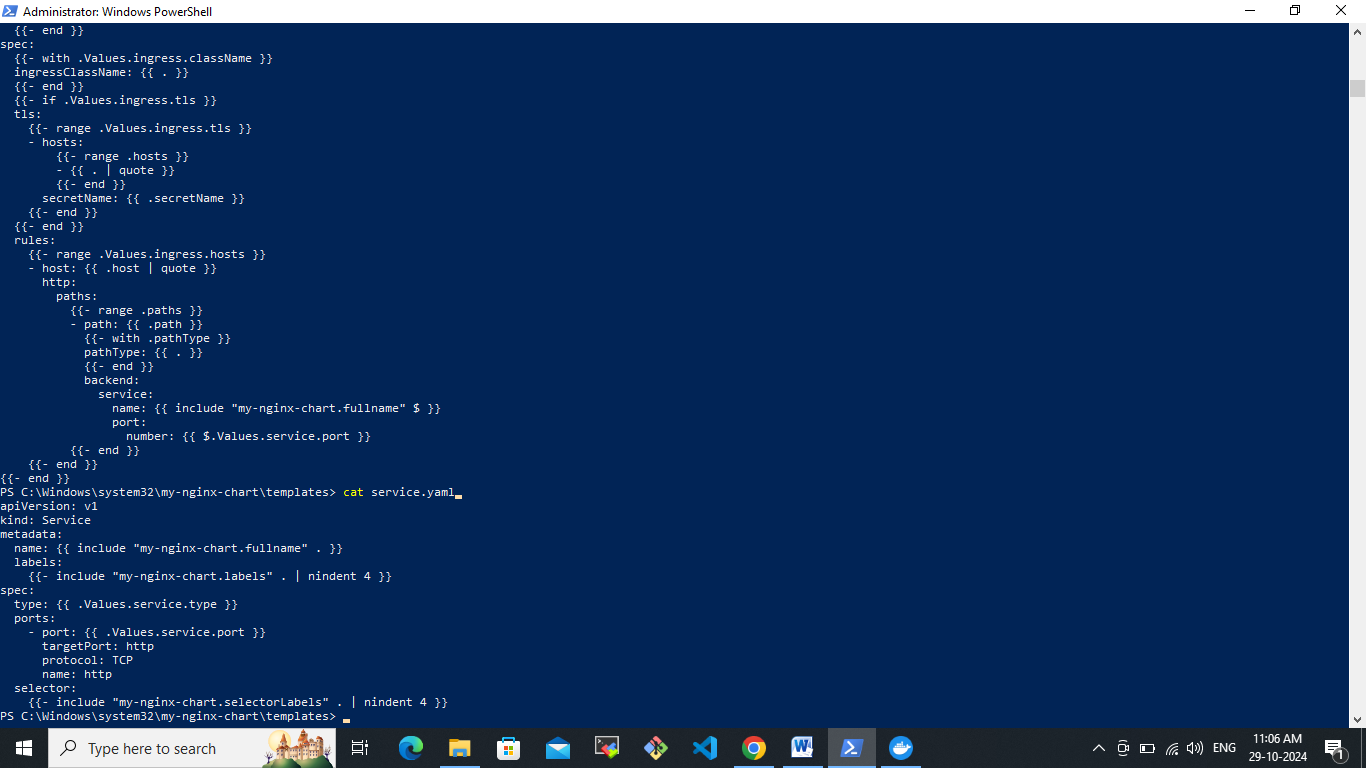
Let’s go with charts .yaml file **has your entire configuration, which is there for your nginix charts right. You can put the chat version how you want to do it. You can change this right.**

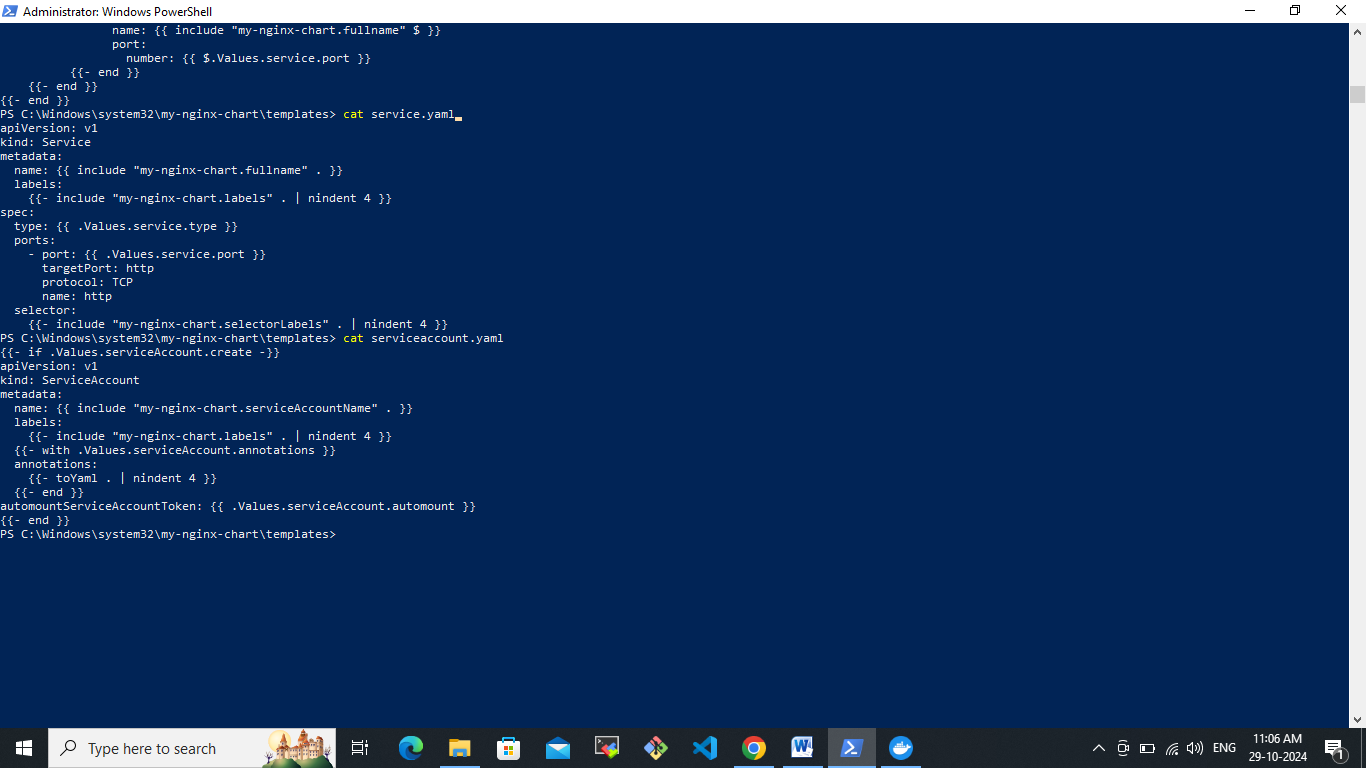
  


HPA- Horizontal pod auto-scaling

When we go with templates you could see what are the files it might have let’s go to templates location.



We are going to deploy the nginx application by using kubernetes helm . First we need to change from the deployments and values.yaml files.

Let’s create deployment.yaml file

We will do nginx deployment using helm.

Deployment.yaml :

apiVersion: apps/v1

kind: Deployment

metadata:

name: {{ include "my-nginx-chart.fullname" . }}

labels:

app: {{ include "my-nginx-chart.name" . }}

spec:

replicas: {{ .Values.replicaCount }}

selector:

matchLabels:

app: {{ include "my-nginx-chart.name" . }}

template:

metadata:

labels:

app: {{ include "my-nginx-chart.name" . }}

spec:

containers:

- name: {{ .Chart.Name }}

image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"

ports:

- containerPort: 80

resources:

requests:

memory: "{{ .Values.resources.requests.memory }}"

cpu: "{{ .Values.resources.requests.cpu }}"

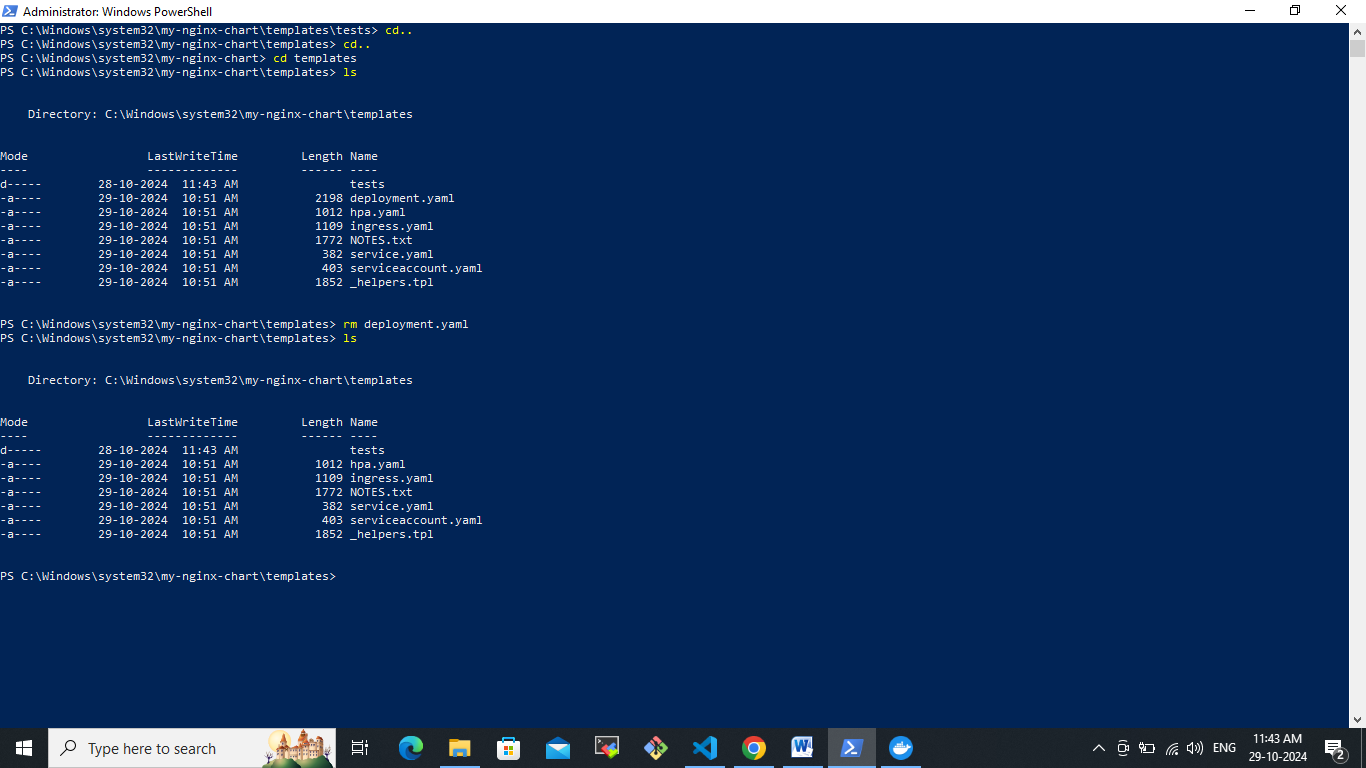
limits:

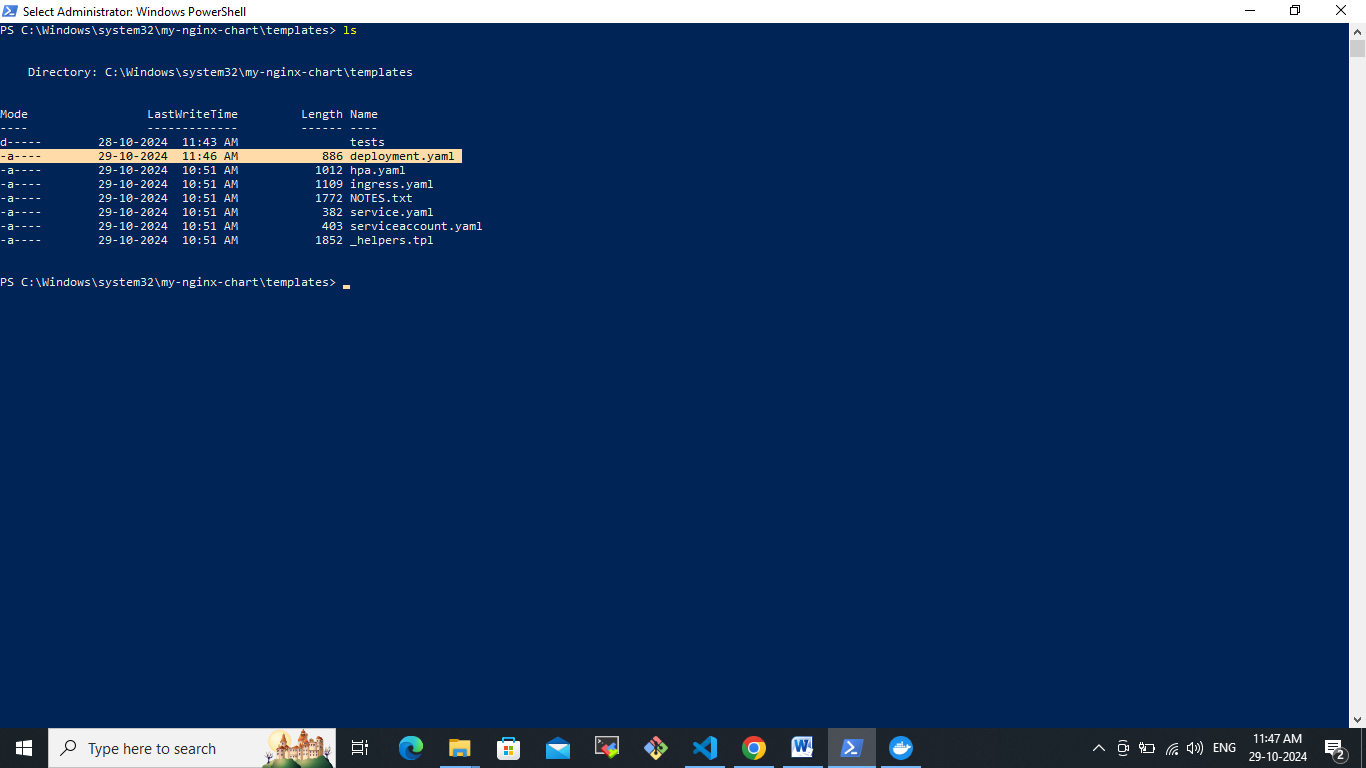
memory: "{{ .Values.resources.limits.memory }}"

cpu: "{{ .Values.resources.limits.cpu }}"

Let’s go to our created nginx-chart in our local right C://windows\\system32\\my-nginx-chart which we created earlier.

Goes to the template location and remove the existing deployment file which was created by default by using command like rm deployment.yaml and create a new file vi deployment.yaml



  
Let’s create Values.yaml file  
Values.yaml

replicaCount: 2

image:

repository: nginx

pullPolicy: IfNotPresent

tag: "1.21.6"

service:

type: ClusterIP

port: 80

resources:

limits:

cpu: 100m

memory: 128Mi

requests:

cpu: 100m

memory: 64Mi

serviceAccount:

create: true

name: ""

ingress:

enabled: false

annotations: {}

hosts:

- host: chart-example.local

paths:

- /

tls: []

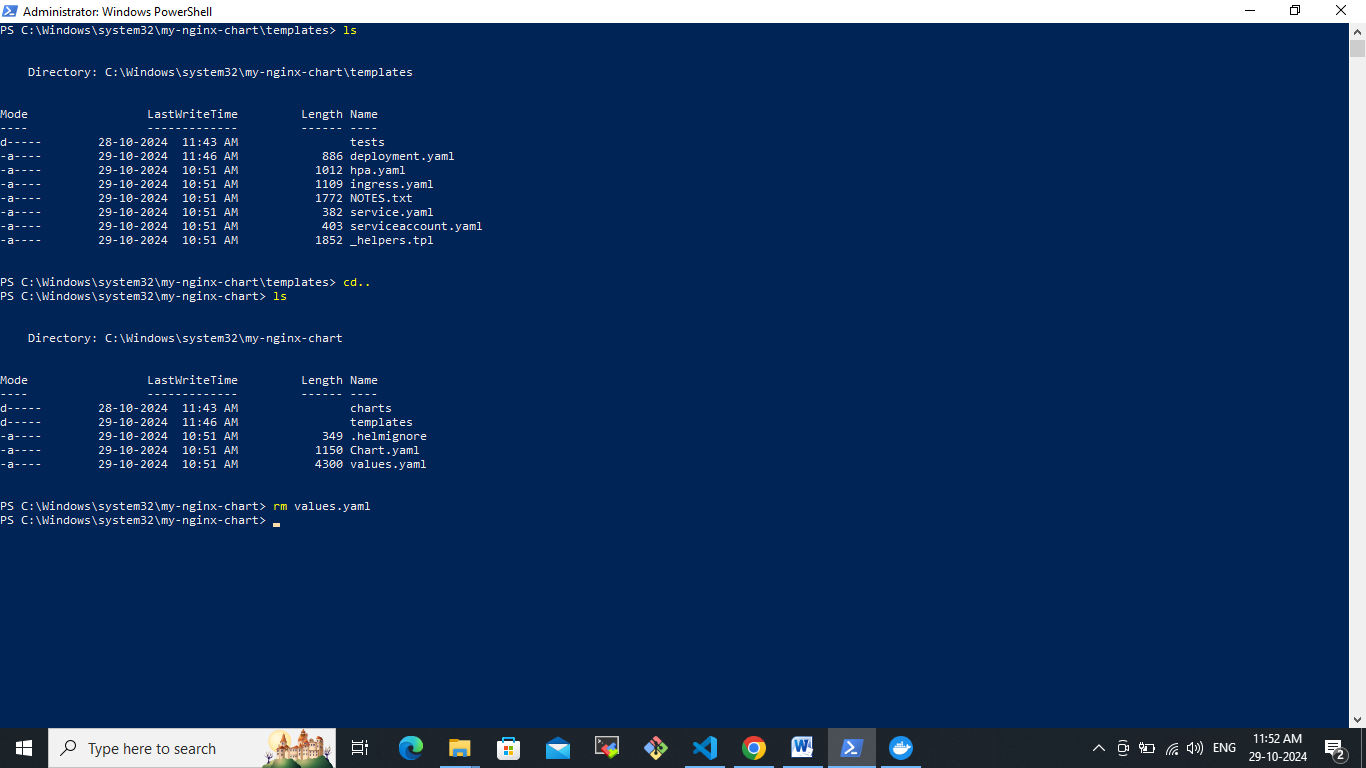
autoscaling:

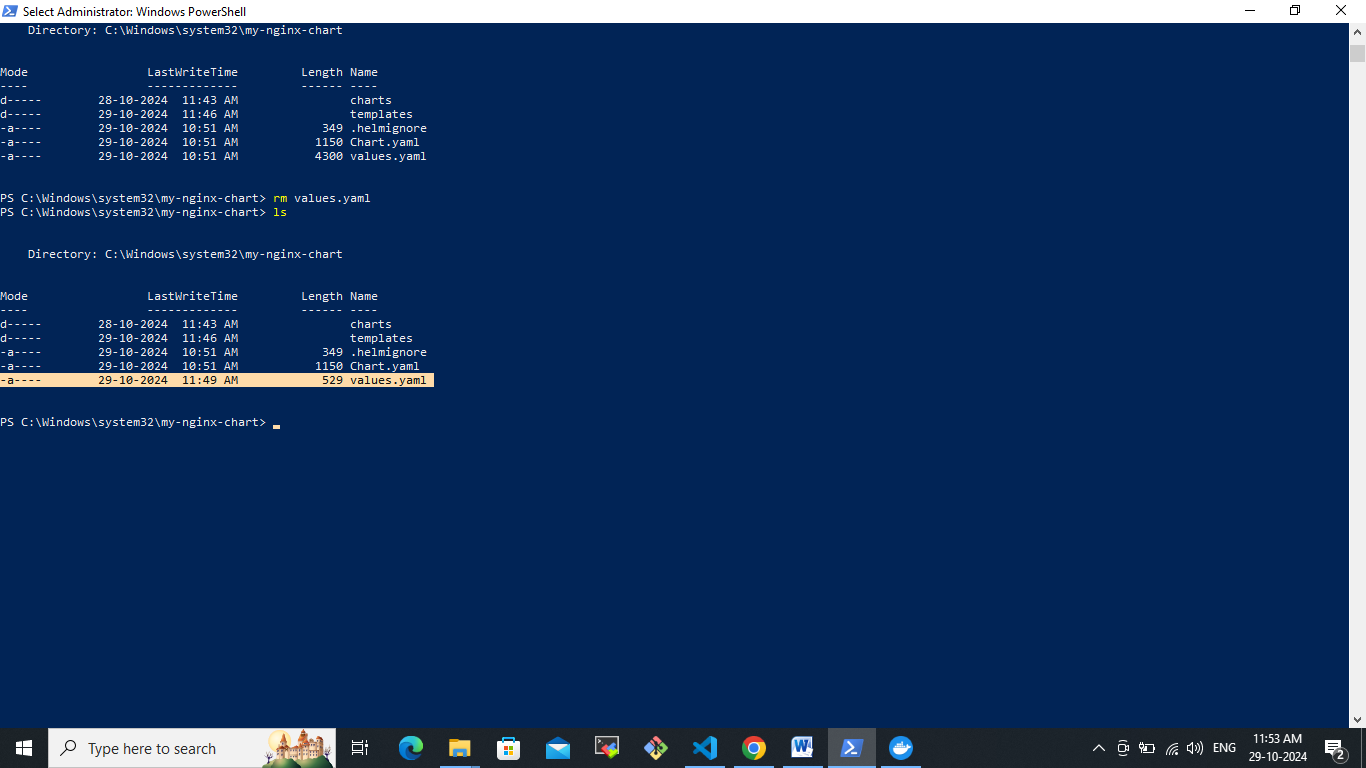
enabled: false

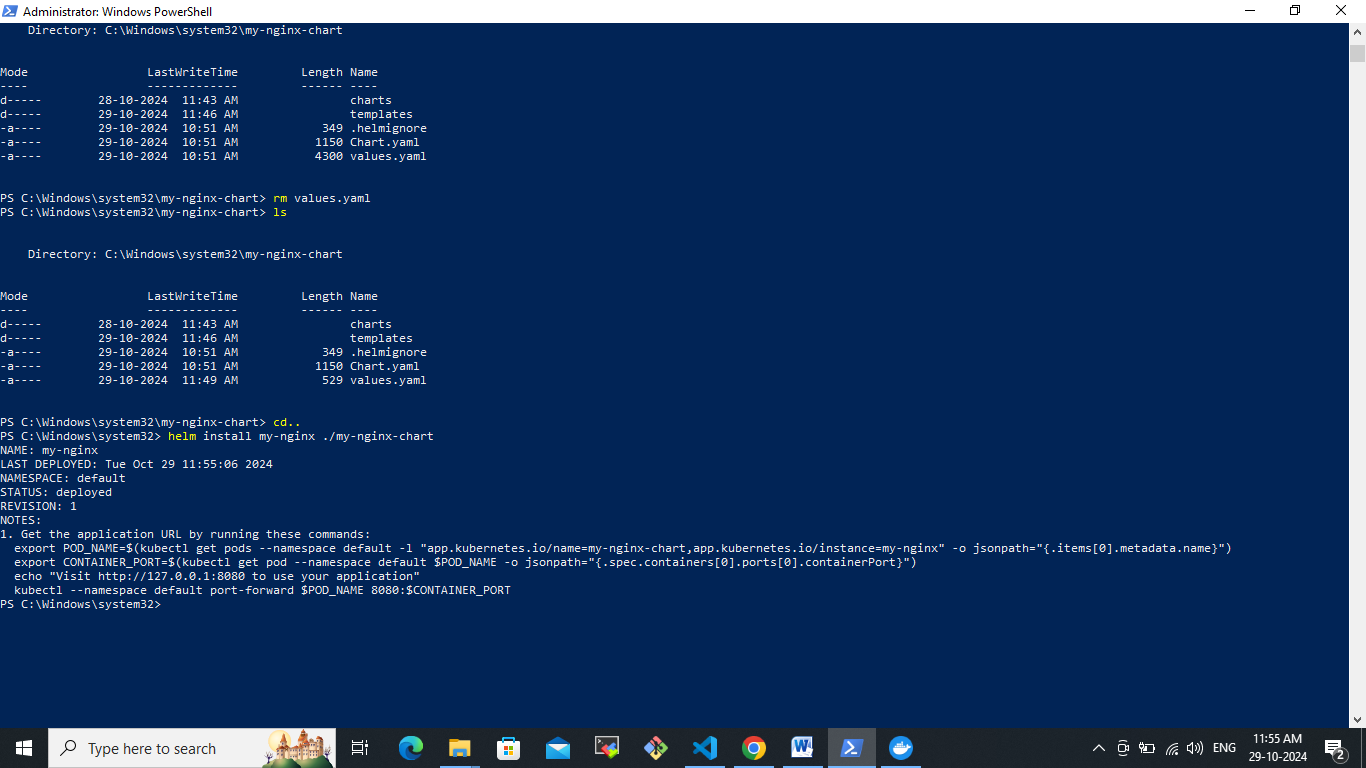
minReplicas: 1

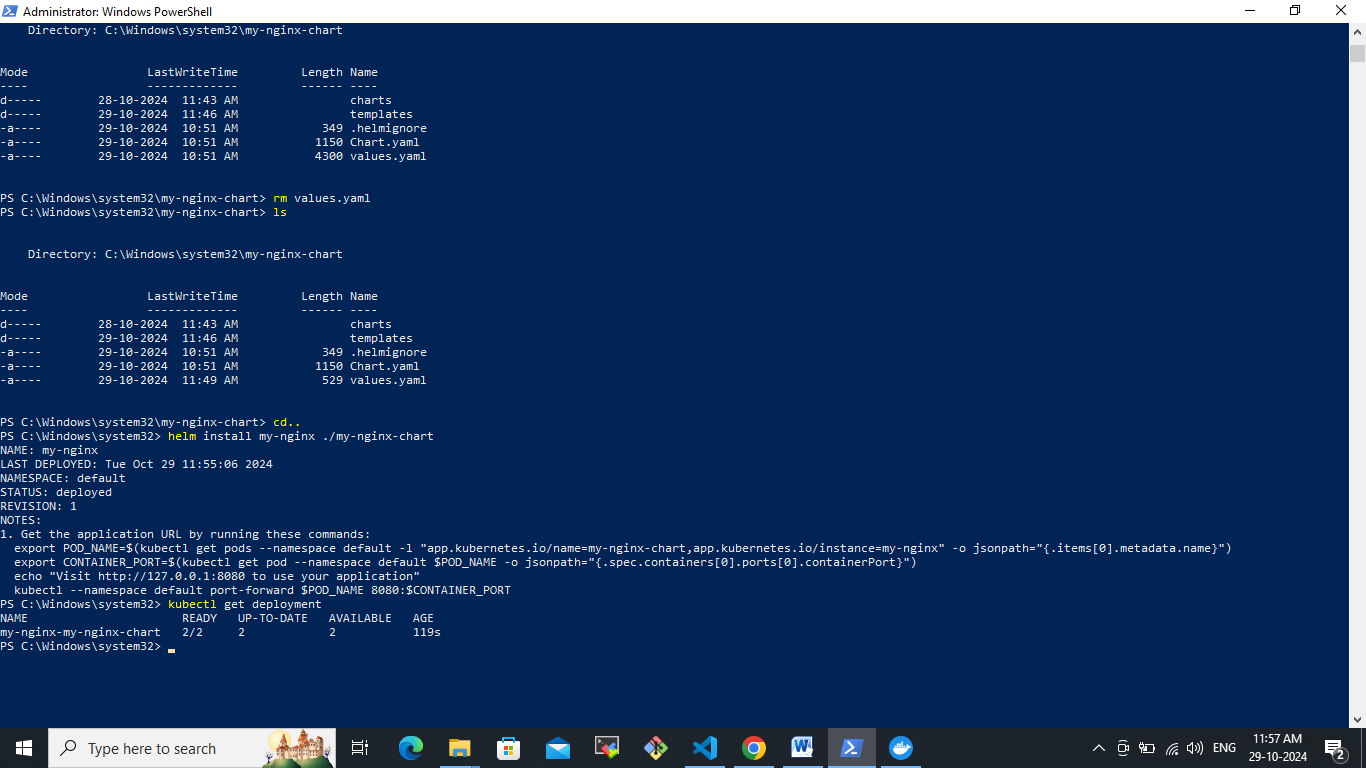
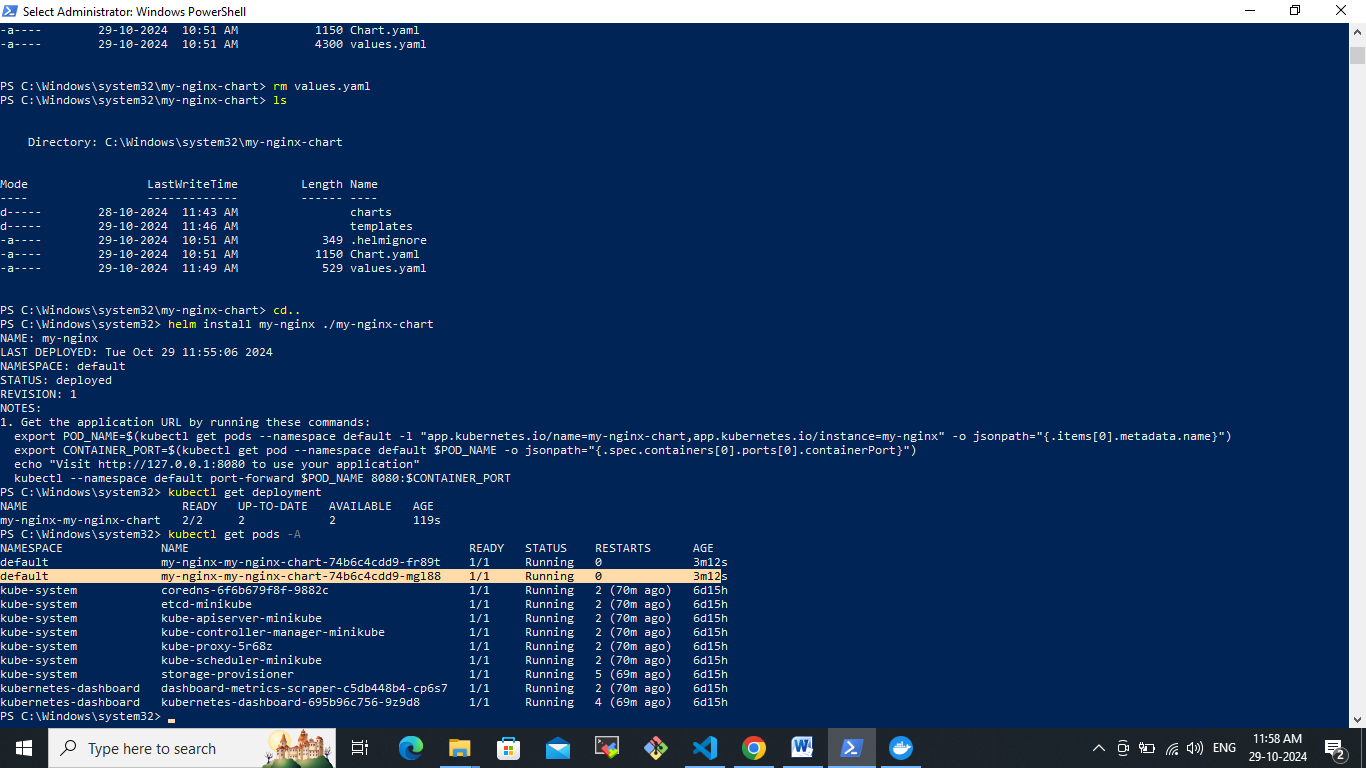
maxReplicas: 10

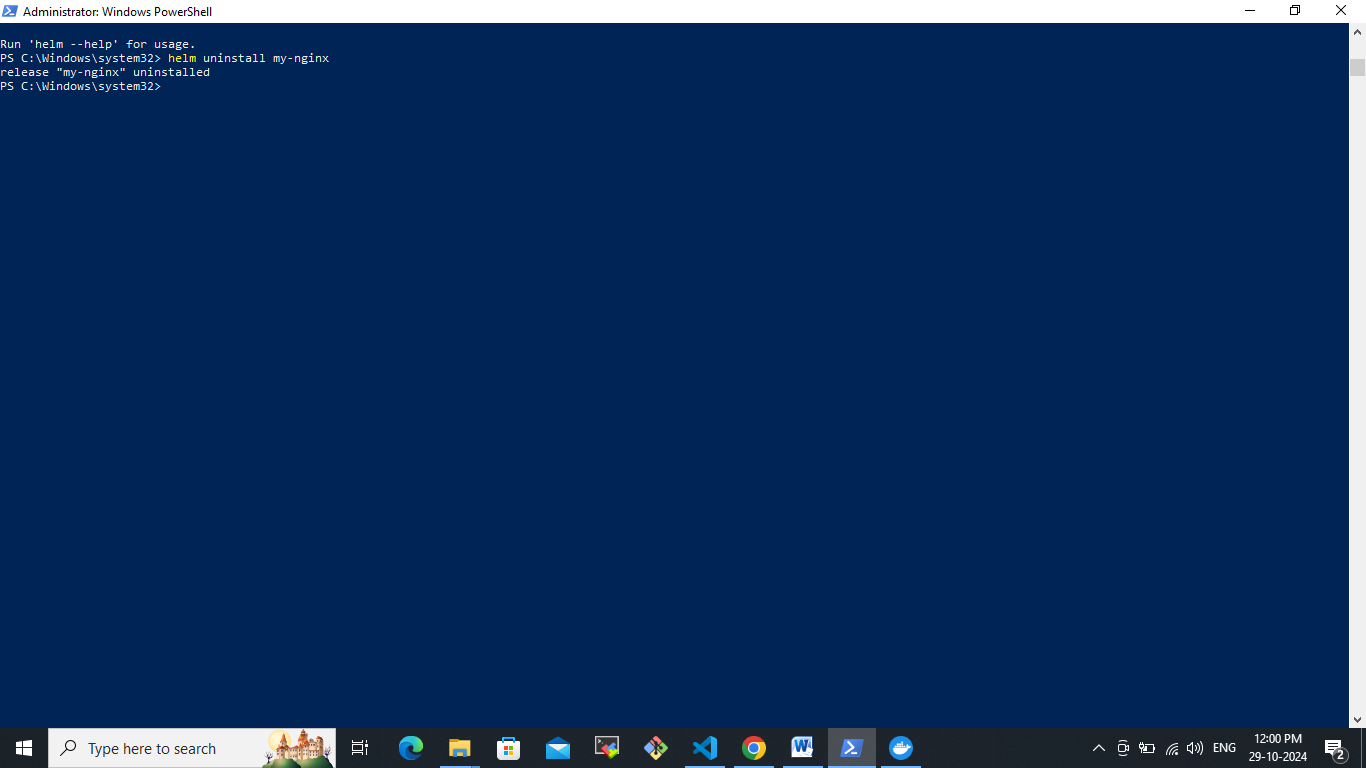
targetCPUUtilizationPercentage: 80

Let me delete the existing values.yaml file and create and added this new one.  


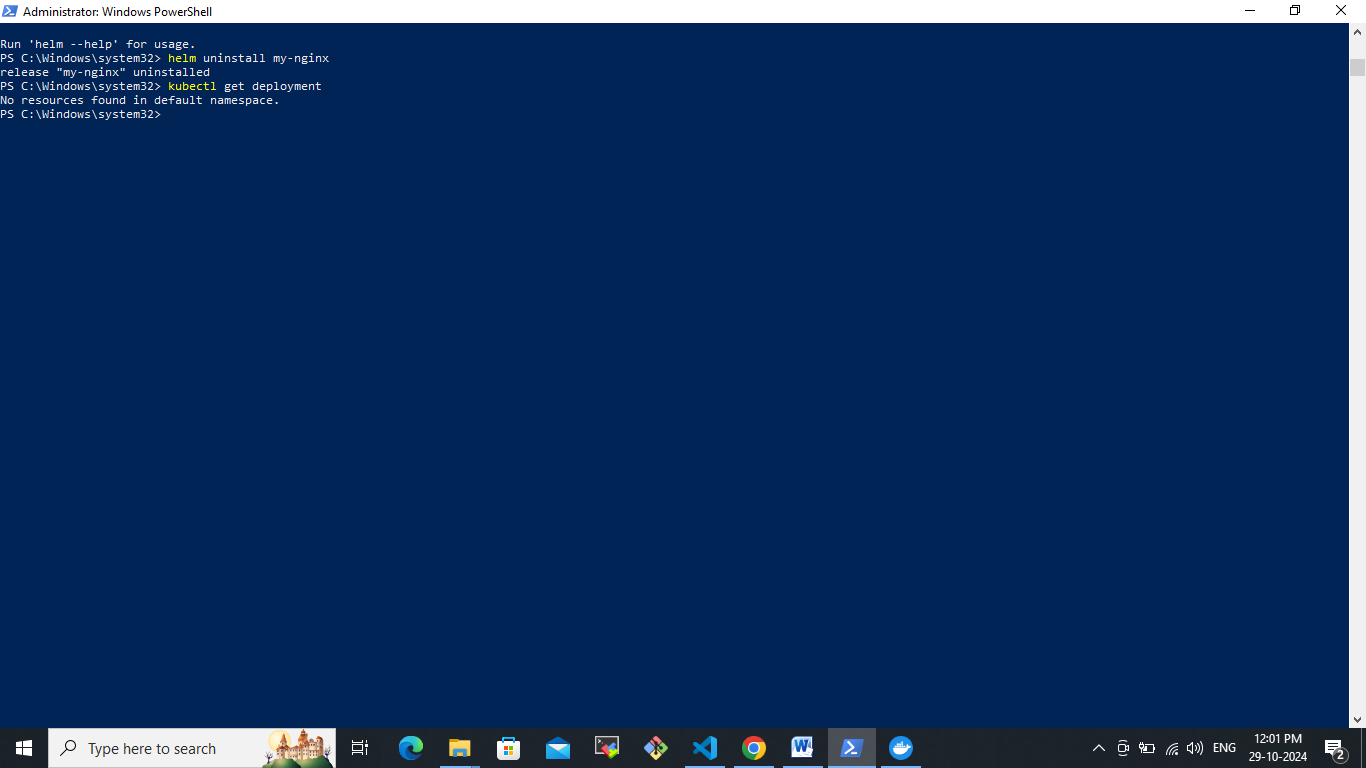
  
now you need to install helm in locally.  
helm install my-nginx ./my-nginx-chart

  
now we have have deployed nginx application by using helm locally.  
let’s check the deployment has been run or not by using this query.  
kubectl get deployement

here 2 replicas are running.  
let’s check the running pod’s  
  
We can simply uninstall by using this app name which we created.



That’s it we have uninstalled it.



Since we have deleted it right.