**Template:**

<https://github.com/zhellDarwin/helmshopapi>

Download this template and change the image name to yours

**Prerequisites:**

1. Need to have aks(azure kubernetes service)
2. Helm installed
3. Template mentioned above

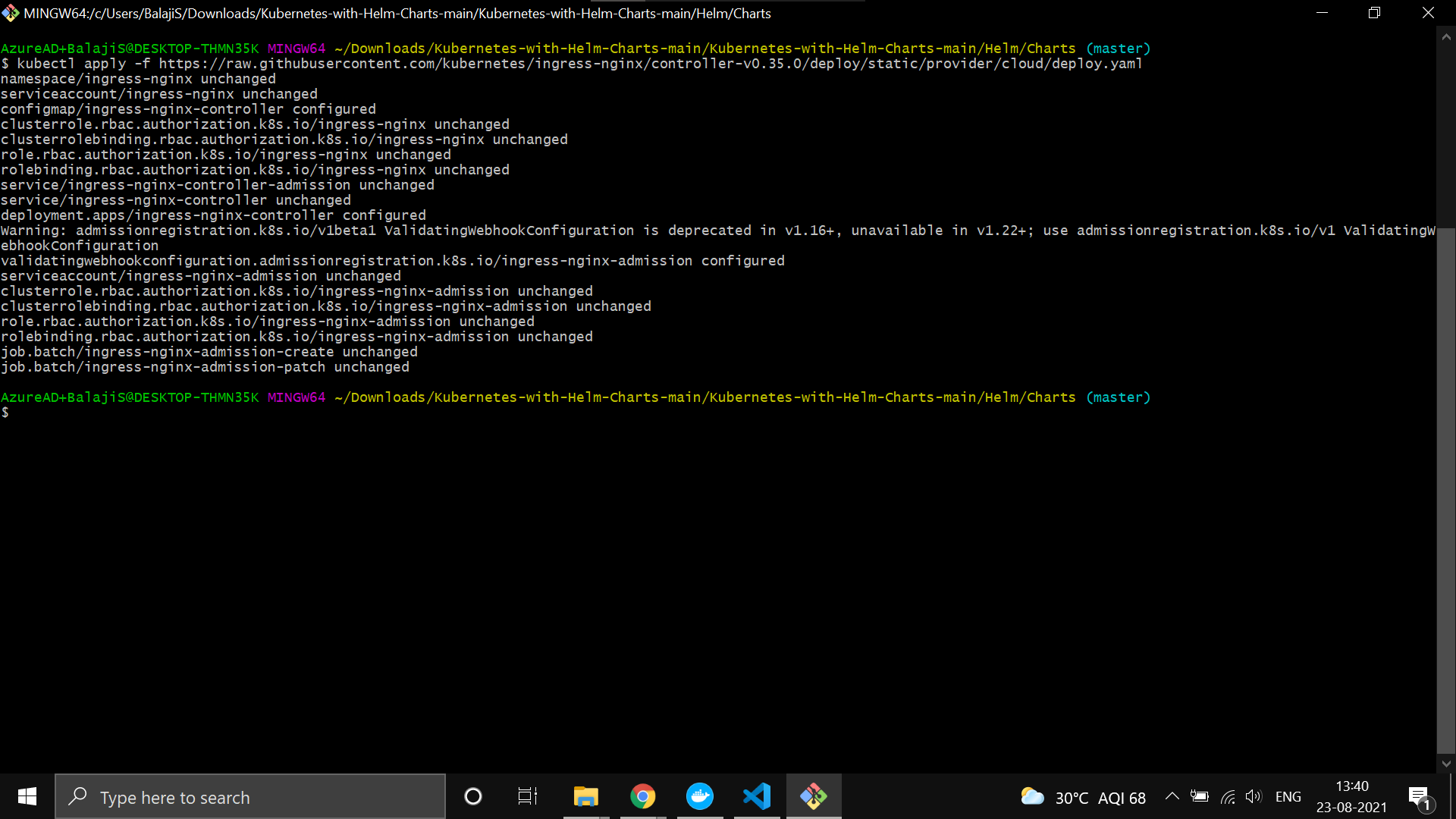
**Initial commands:**

* az login
* az account set --subscription <subscription id>
* az aks get-credentials --resource-group <resource group name>--name <kubernetes service name>

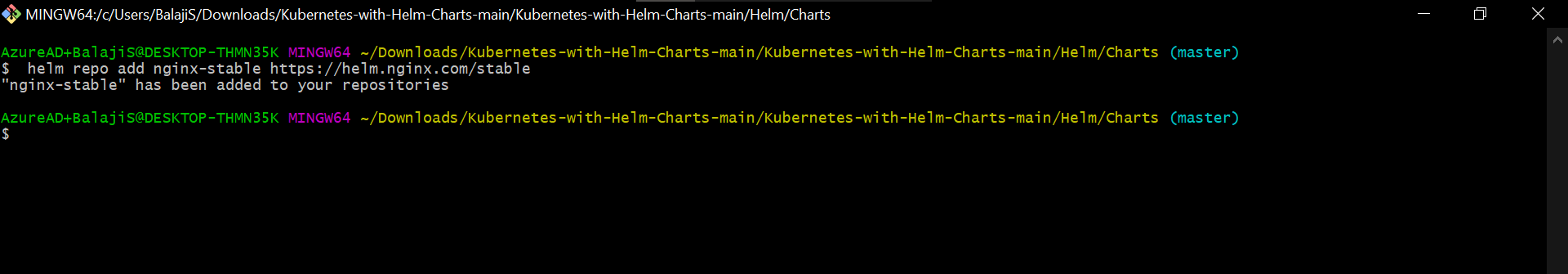
Open the command prompt and do the above commands. Change the underlined area in the commands to your aks configuration.

**Install the nginx chart to the kubectl, without this the nginx ingress will not function.**

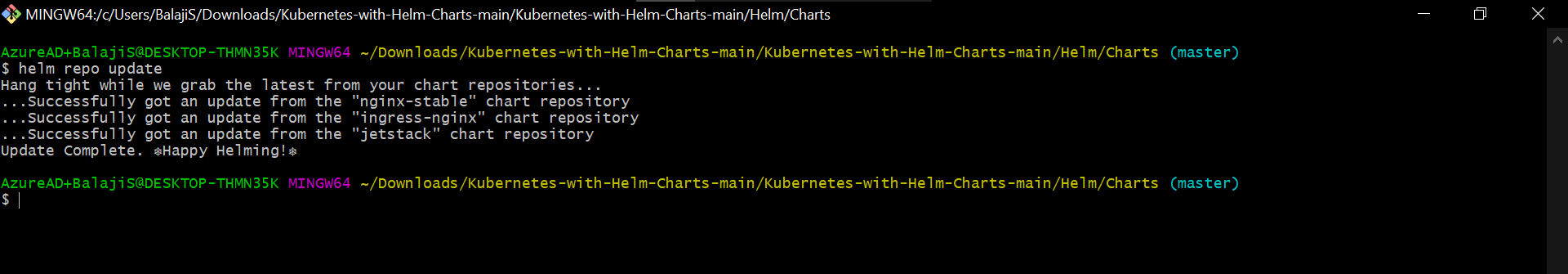
* kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.35.0/deploy/static/provider/cloud/deploy.yaml



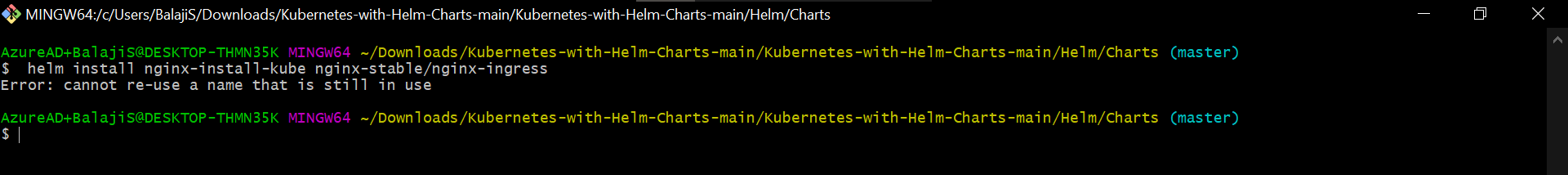
* helm repo add nginx-stable https://helm.nginx.com/stable



* helm repo update



* helm install nginx-install-kube nginx-stable/nginx-ingress

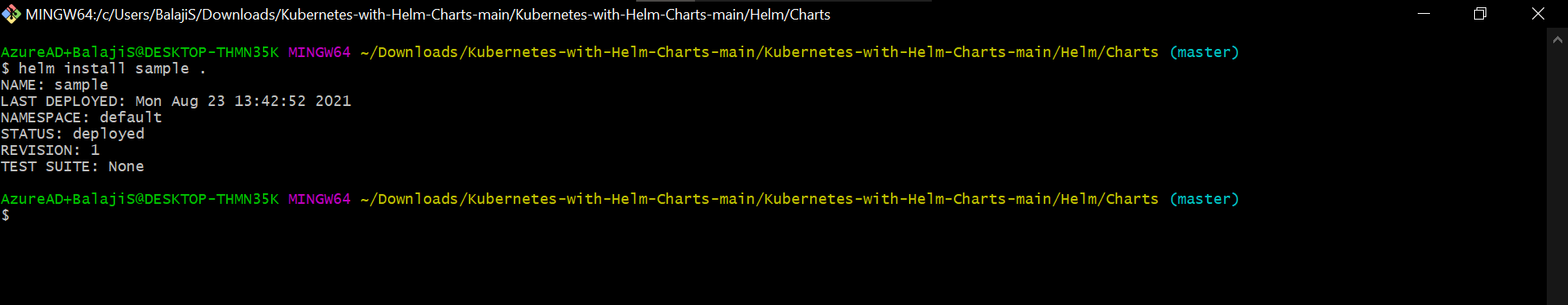


I already installed it so it shows an error.

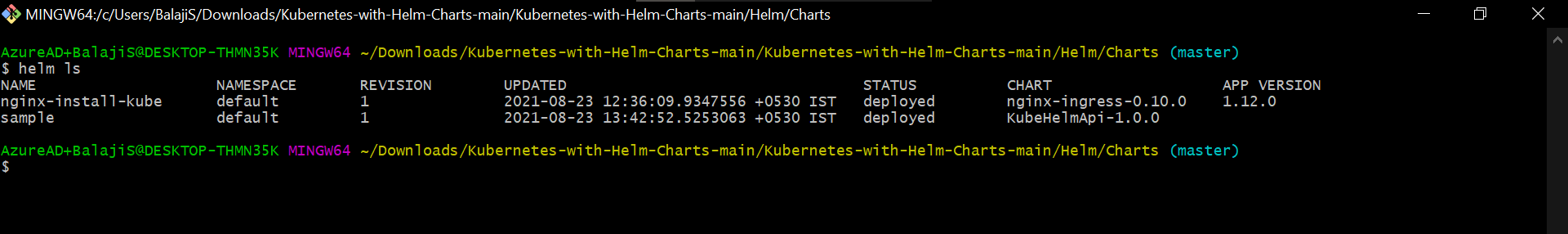
After running the above command go to the downloaded template inside the charts folder.

Do the following command.

**helm install <some name for your api> .** (dot is important)

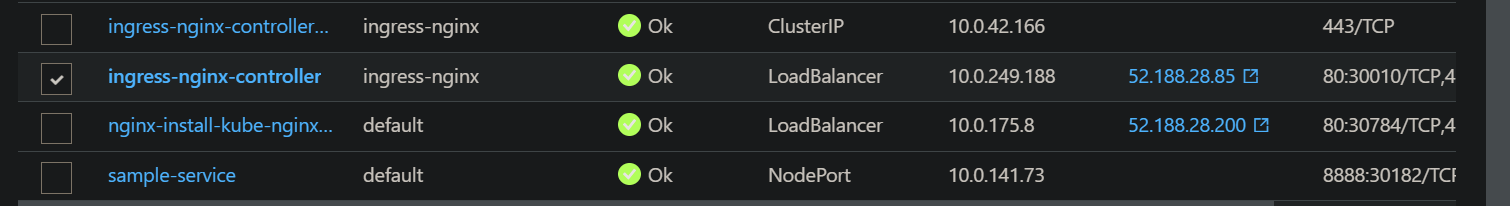
****

helm ls - To show the installed charts.



kubectl get ingress - To get the ingress ip of the installed chart.

Or go to the services and ingress in your kubernetes service and find your external ip.



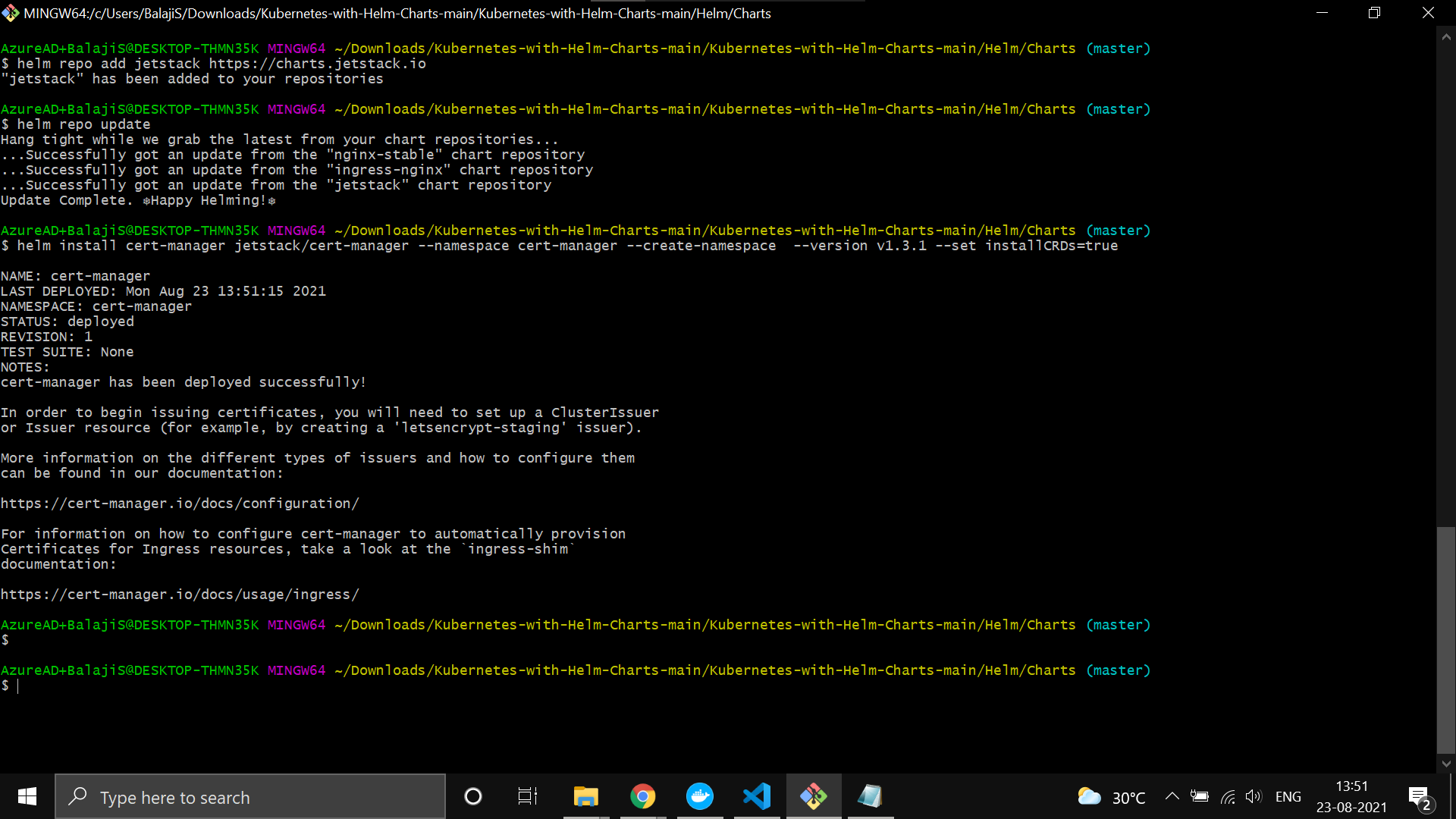
ingress-nginx-controller external ip is our application ip.

**To get ssl certification:**

Installing the cert-manager:

* helm repo add jetstack https://charts.jetstack.io
* helm repo update
* helm install cert-manager jetstack/cert-manager --namespace cert-manager \

--create-namespace --version v1.3.1 --set installCRDs=true



staging.yaml

apiVersion: cert-manager.io/v1

kind: ClusterIssuer

metadata:

name: letsencrypt-staging

spec:

acme:

# You must replace this email address with your own.

# Let's Encrypt will use this to contact you about expiring

# certificates, and issues related to your account.

email: user@email.com

server: https://acme-staging-v02.api.letsencrypt.org/directory

privateKeySecretRef:

# Secret resource that will be used to store the account's private key.

name: letsencrypt-staging-ssl

# Add a single challenge solver, HTTP01 using nginx

solvers:

- http01:

ingress:

class: nginx

production.yaml

apiVersion: cert-manager.io/v1

kind: ClusterIssuer

metadata:

name: letsencrypt-staging

spec:

acme:

# You must replace this email address with your own.

# Let's Encrypt will use this to contact you about expiring

# certificates, and issues related to your account.

email: user@email.com

server: https://acme-staging-v02.api.letsencrypt.org/directory

privateKeySecretRef:

# Secret resource that will be used to store the account's private key.

name: letsencrypt-staging-ssl

# Add a single challenge solver, HTTP01 using nginx

solvers:

- http01:

ingress:

class: nginx

certificates.yaml

apiVersion: cert-manager.io/v1alpha2

kind: Certificate

metadata:

namespace: default

name: hello-certs

spec:

secretName: hello-certs

issuerRef:

name: letsencrypt-prod

kind: ClusterIssuer

dnsNames:

- hello.demo.com #dns you bought

kubectl apply -f <filename>

<https://dev.to/csgeek/kubernetes-ssl-letsencrypt-bmf>

Verify this for ssl certificate.

**Issues:**

* Nginx-ingress-controller-failed-calling-webhook

<https://stackoverflow.com/questions/61616203/nginx-ingress-controller-failed-calling-webhook>

**Monitoring:**

<https://k21academy.com/docker-kubernetes/prometheus-grafana-monitoring/>

**Dotnet logging to docker logs**

<https://stackoverflow.com/questions/49167186/asp-net-core-net-core-console-app-logging-in-docker-container>

* helm install nginx-install-kube nginx-stable/nginx-ingress

helm install dashboardbuilderui-nginx nginx-stable/nginx-ingress --set controller.ingressClass=first --namespace dashboardbuilderui --set controller.replicaCount=1 --set rbac.create=false

helm install formbuilderui-nginx nginx-stable/nginx-ingress --set controller.ingressClass=second --namespace formbuilderui --set controller.replicaCount=1 --set rbac.create=false