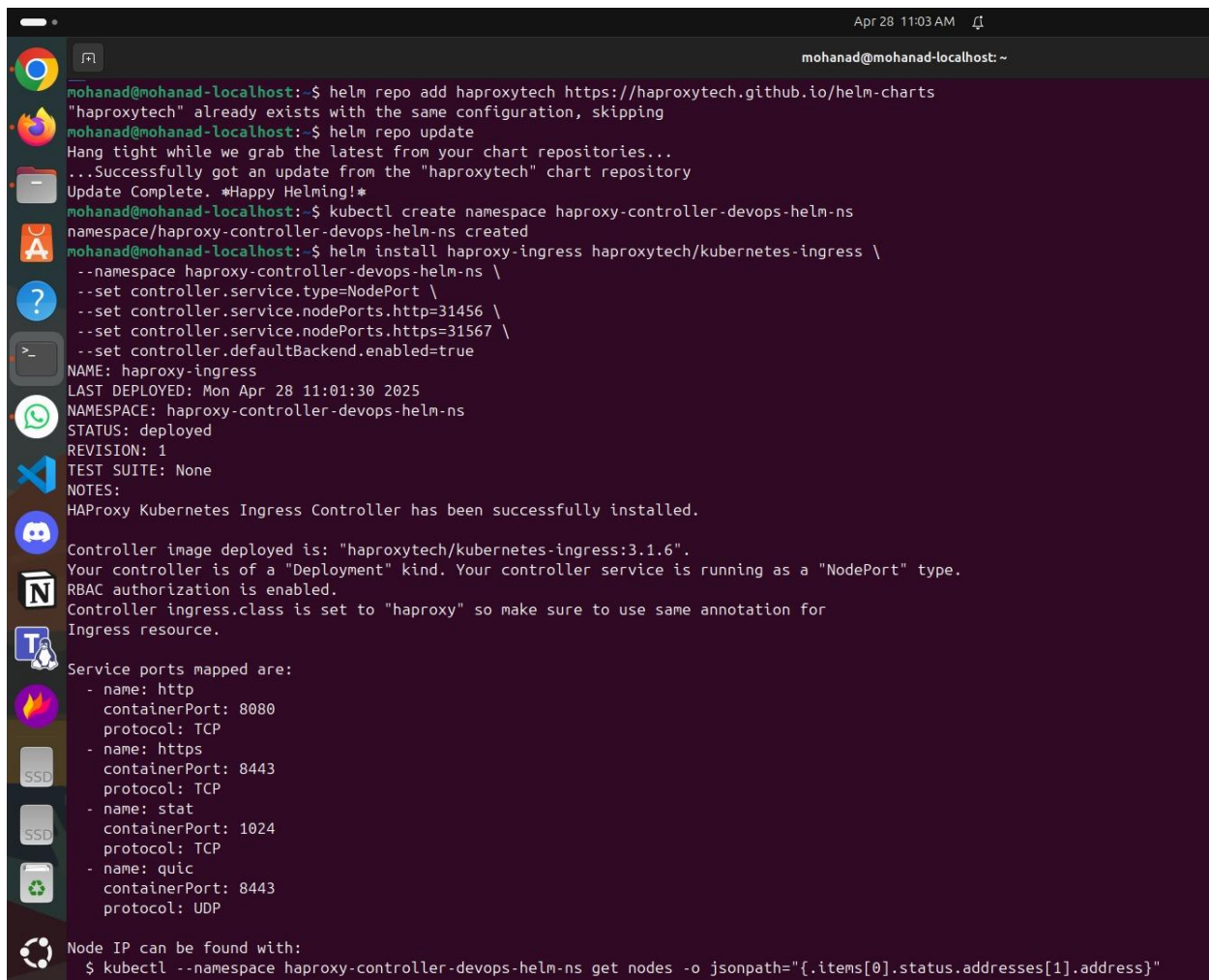


Mohanad Mahmoud Sayed

Helm Task

- ✓ Firstly, download the haproxy helm chart, and configure the ports of http and https as required in lab5:



```
Apr 28 11:03 AM
mohanad@mohanad-localhost: ~

mohanad@mohanad-localhost:~$ helm repo add haproxytech https://haproxytech.github.io/helm-charts
"haproxytech" already exists with the same configuration, skipping
mohanad@mohanad-localhost:~$ helm repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "haproxytech" chart repository
Update Complete. *Happy Helming!*
mohanad@mohanad-localhost:~$ kubectl create namespace haproxy-controller-devops-helm-ns
namespace/haproxy-controller-devops-helm-ns created
mohanad@mohanad-localhost:~$ helm install haproxy-ingress haproxytech/kubernetes-ingress \
--namespace haproxy-controller-devops-helm-ns \
--set controller.service.type=NodePort \
--set controller.service.nodePorts.http=31456 \
--set controller.service.nodePorts.https=31567 \
--set controller.defaultBackend.enabled=true
NAME: haproxy-ingress
LAST DEPLOYED: Mon Apr 28 11:01:30 2025
NAMESPACE: haproxy-controller-devops-helm-ns
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
HAProxy Kubernetes Ingress Controller has been successfully installed.

Controller image deployed is: "haproxytech/kubernetes-ingress:3.1.6".
Your controller is of a "Deployment" kind. Your controller service is running as a "NodePort" type.
RBAC authorization is enabled.
Controller ingress.class is set to "haproxy" so make sure to use same annotation for
Ingress resource.

Service ports mapped are:
- name: http
  containerPort: 8080
  protocol: TCP
- name: https
  containerPort: 8443
  protocol: TCP
- name: stat
  containerPort: 1024
  protocol: TCP
- name: quic
  containerPort: 8443
  protocol: UDP

Node IP can be found with:
$ kubectl --namespace haproxy-controller-devops-helm-ns get nodes -o jsonpath="{.items[0].status.addresses[1].address}"
```

```
Apr 28 11:04 AM
mohanad@mohanad-localhost: ~

- name: https
  containerPort: 8443
  protocol: TCP
- name: stat
  containerPort: 1024
  protocol: TCP
- name: quic
  containerPort: 8443
  protocol: UDP

Node IP can be found with:
$ kubectl --namespace haproxy-controller-devops-helm-ns get nodes -o jsonpath="{.items[0].status.addresses[1].address}"

The following ingress resource routes traffic to pods that match the following:
* service name: web
* client's Host header: webdemo.com
* path begins with /

---
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: web-ingress
  namespace: default
  annotations:
    ingress.class: "haproxy"
spec:
  rules:
    - host: webdemo.com
      http:
        paths:
          - path: /
            backend:
              serviceName: web
              servicePort: 80

In case that you are using multi-ingress controller environment, make sure to use ingress.class annotation and match it
with helm chart option controller.ingressClass.

For more examples and up to date documentation, please visit:
* Helm chart documentation: https://github.com/haproxytech/helm-charts/tree/main/kubernetes-ingress
* Controller documentation: https://www.haproxy.com/documentation/kubernetes/latest/
* Annotation reference: https://github.com/haproxytech/kubernetes-ingress/tree/master/documentation
* Image parameters reference: https://github.com/haproxytech/kubernetes-ingress/blob/master/documentation/controller.md
mohanad@mohanad-localhost: ~
```

```
mohanad@mohanad-localhost: ~$ kubectl get pods -n haproxy-controller-devops-helm-ns
NAME                                READY   STATUS    RESTARTS   AGE
haproxy-ingress-kubernetes-ingress-b87449cc7-7d85t  1/1     Running   0           4m10s
haproxy-ingress-kubernetes-ingress-b87449cc7-qxrzd  1/1     Running   0           4m10s
haproxy-ingress-kubernetes-ingress-crdjob-1-gmckx    0/1     Completed 0           4m10s
mohanad@mohanad-localhost: ~$ kubectl get svc -n haproxy-controller-devops-helm-ns
NAME                                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)                                     AGE
haproxy-ingress-kubernetes-ingress  NodePort    10.96.124.48 <none>        80:31456/TCP,443:31567/TCP,443:31567/UDP,1024:31485/TCP 4m34s
mohanad@mohanad-localhost: ~
```

Output&Result:

Apr 28 11:08 AM

Statistics Report for HAP: x

Not Secure http://192.168.49.2:31485

HAProxy version 3.1.6-d929ca2, released 2025/03/20

Statistics Report for pid 147

> General process information

pid = 147 (process #1, nproc = 1, nbthread = 16)
uptime = 0d 0h03m32s; warnings = 0
system limits: memmax = 9216 MB; ulimit-n = 144171
maxsock = 144171; maxconn = 72000; reached = 0; maxpipes = 0
current conns = 2; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.271 kbps
Running tasks: 0/135 (0 nice); idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance
backup UP
backup UP, going down
backup DOWN, going up
not checked
Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope:
- [Hide DOWN servers](#)
- [Disable refresh](#)
- [Refresh now](#)
- [CSV export](#)
- [JSON export \(schema\)](#)

External resources:

- [Primary site](#)
- [Updates \(v3.1\)](#)
- [Online manual](#)

healthz																															
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtle	
Frontend				1	2	-	0	6	72 000	44			4 136	6 600	0	0	0					OPEN									

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtle	
Frontend				0	1	-	1	3	72 000	1			948	405	0	0	0					OPEN									

https																															
	Queue			Session rate			Sessions					Bytes		Denied		Errors			Warnings		Server										
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtle	
Frontend				0	0	-	0	0	72 000	0			0	0	0	0	0					OPEN									

stats																															
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server									
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtle	
Frontend				0	1	-	1	3	72 000	1			309	208 982	0	0	0					OPEN									

haproxy-controller-devops-helm-nls_default-local-service_http																														
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntime	Thrtle
SRV_1	0	0	-	0	2		0	1	-	3	3	30s	948	405		0		0	0	0	0	3m32s UP	L4OK in 0ms	1/1	Y	-	0	0	0s	-
SRV_2	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_3	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_4	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_5	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_6	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_7	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_8	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_9	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_10	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_11	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_12	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-
SRV_13	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	3m32s MAINT		1/1	Y	-	0	0	3m32s	-