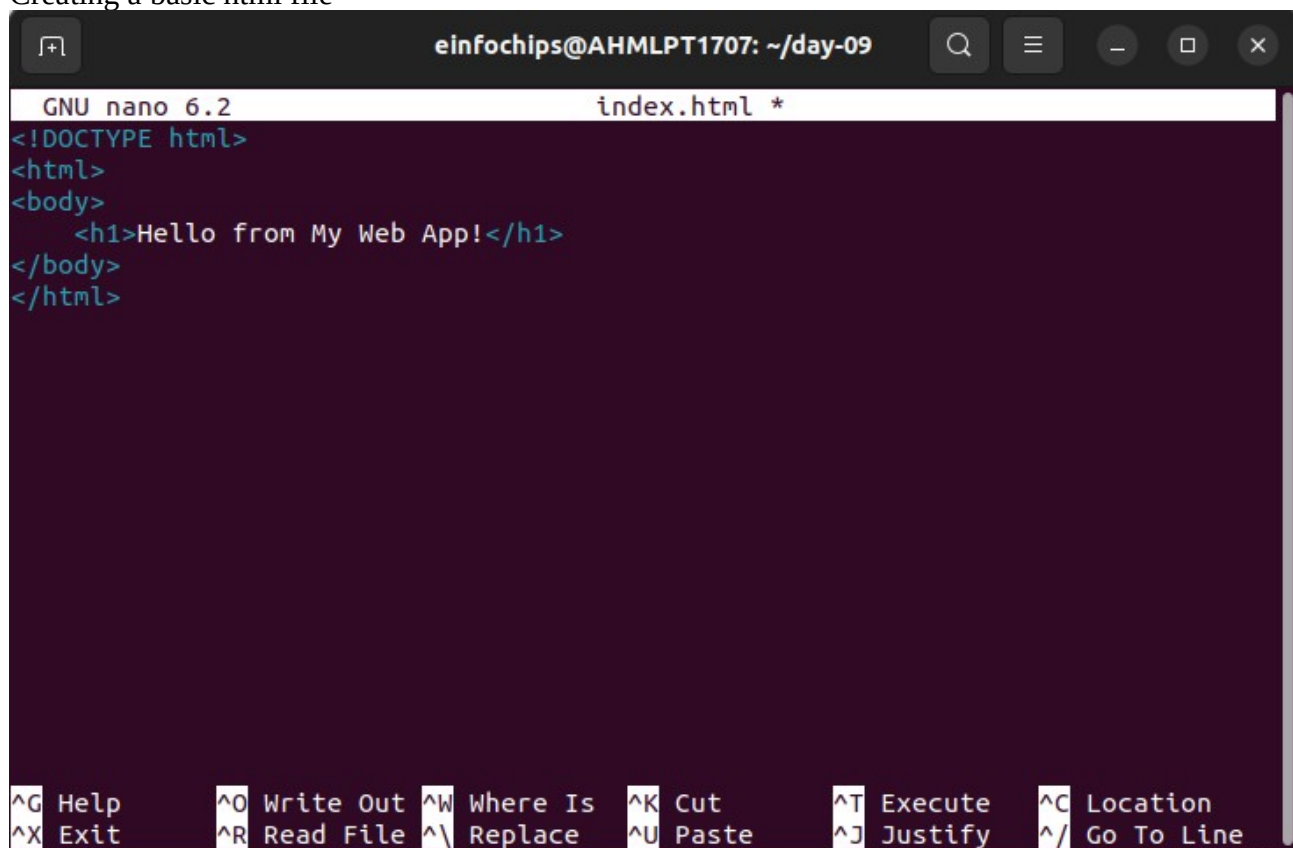


Day 09- Kubernetes ingress, URL rewriting, sticky sessions, and autoscaling.

1: Setting Up the Kubernetes Cluster and Static Web App

```
einfochips@AHMLPT1707:~$ mkdir day-09
einfochips@AHMLPT1707:~$ minikube start
🐹 minikube v1.33.1 on Ubuntu 22.04
🌟 Using the docker driver based on existing profile
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🌱 Pulling base image v0.0.44 ...
🔄 Updating the running docker "minikube" container ...
🔧 Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
🔍 Verifying Kubernetes components...
   ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🏁 Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
einfochips@AHMLPT1707:~$
```

Creating a basic html file



```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 index.html *
<!DOCTYPE html>
<html>
<body>
  <h1>Hello from My Web App!</h1>
</body>
</html>
```

creating docker file

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 Dockerfile *
# Use the official Nginx base image
FROM nginx:latest

# Copy the custom HTML file to the appropriate location
COPY index.html /usr/share/nginx/html/

# Expose port 80
EXPOSE 80

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Building the dockerhub image

```
einfochips@AHMLPT1707: ~/day-09
"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
einfochips@AHMLPT1707:~/day-09$ sudo docker build -t yashmahi04/web-app-09 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.

           Install the buildx component to build images with BuildKit:
           https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.072kB
Step 1/3 : FROM nginx:latest
----> ffffffc90d343
Step 2/3 : COPY index.html /usr/share/nginx/html/
----> fb240fcaef32
Step 3/3 : EXPOSE 80
----> Running in 6cab24493749
Removing intermediate container 6cab24493749
----> 3052b9a08d08
Successfully built 3052b9a08d08
Successfully tagged yashmahi04/web-app-09:latest
einfochips@AHMLPT1707:~/day-09$
```

Pushing the image to the docker hub

```
einfochips@AHMLPT1707: ~/day-09
Step 1/3 : FROM nginx:latest
---> fffffc90d343
Step 2/3 : COPY index.html /usr/share/nginx/html/
---> fb240fcaef32
Step 3/3 : EXPOSE 80
---> Running in 6cab24493749
Removing intermediate container 6cab24493749
---> 3052b9a08d08
Successfully built 3052b9a08d08
Successfully tagged yashmahi04/web-app-09:latest
einfochips@AHMLPT1707:~/day-09$ sudo docker push yashmahi04/web-app-09
Using default tag: latest
The push refers to repository [docker.io/yashmahi04/web-app-09]
4325decc1581: Pushed
56b6d3be75f9: Mounted from library/nginx
0c6c257920c8: Mounted from library/nginx
92d0d4e97019: Mounted from library/nginx
7190c87a0e8a: Mounted from library/nginx
933a3ce2c78a: Mounted from library/nginx
32cfaf91376f: Mounted from library/nginx
32148f9f6c5a: Mounted from library/postgres
latest: digest: sha256:d8d7bdb90d9068b5b94e46ac0dc7839306ea9c2c7cb3860bea56a8ca3
e4f7e59 size: 1985
einfochips@AHMLPT1707:~/day-09$
```

Kubernetes Deployment:
creating deployment.yaml

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 deployment.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: static-web-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: static-web
  template:
    metadata:
      labels:
        app: static-web
    spec:
      containers:
        - name: nginx
          image: <your-docker-username>/<image-name>
          ports:
            - containerPort: 80

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```


creating service.yaml

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 service.yaml
apiVersion: v1
kind: Service
metadata:
  name: static-web-service
spec:
  selector:
    app: static-web
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: NodePort

[ Wrote 12 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Applying the deployment.yaml and service.yaml

```
einfochips@AHMLPT1707: ~/day-09
---> 3052b9a08d08
Successfully built 3052b9a08d08
Successfully tagged yashmahi04/web-app-09:latest
einfochips@AHMLPT1707:~/day-09$ sudo docker push yashmahi04/web-app-09
Using default tag: latest
The push refers to repository [docker.io/yashmahi04/web-app-09]
4325decc1581: Pushed
56b6d3be75f9: Mounted from library/nginx
0c6c257920c8: Mounted from library/nginx
92d0d4e97019: Mounted from library/nginx
7190c87a0e8a: Mounted from library/nginx
933a3ce2c78a: Mounted from library/nginx
32cfaf91376f: Mounted from library/nginx
32148f9f6c5a: Mounted from library/postgres
latest: digest: sha256:d8d7bdb90d9068b5b94e46ac0dc7839306ea9c2c7cb3860bea56a8ca3e4f7e59 size: 1985
einfochips@AHMLPT1707:~/day-09$ nano deployment.yaml
einfochips@AHMLPT1707:~/day-09$ nano deployment.yaml
einfochips@AHMLPT1707:~/day-09$ nano service.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f deployment.yaml
deployment.apps/static-web-deployment created
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f service.yaml
service/static-web-service created
einfochips@AHMLPT1707:~/day-09$
```

2: Configuring Ingress Networking

Tried to install the ingress using the command
`minikube start --addons=ingress`

Error occurred

```
einfochips@AHMLPT1707: ~/day-09
einfochips@AHMLPT1707:~/day-09$ nano service.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f deployment.yaml
deployment.apps/static-web-deployment created
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f service.yaml
service/static-web-service created
einfochips@AHMLPT1707:~/day-09$ minikube start --addons=ingress
🌟 minikube v1.33.1 on Ubuntu 22.04
🌟 Using the docker driver based on existing profile
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🌐 Pulling base image v0.0.44 ...
🔧 Updating the running docker "minikube" container ...
🔧 Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
🔧 Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  ■ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
🔧 Verifying ingress addon...
❗ Enabling 'ingress' returned an error: running callbacks: [waiting for app.kubernetes.io/name=ingress-nginx pods: context deadline exceeded]
🌟 Enabled addons: storage-provisioner, default-storageclass
🌟 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
einfochips@AHMLPT1707:~/day-09$ minikube addons enable ingress
```

In order to solve this we have to enable the ingress
`minikube addons enable ingress`

```
einfochips@AHMLPT1707: ~/day-09
space by default
einfochips@AHMLPT1707:~/day-09$ minikube addons enable ingress
💡 ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  ■ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
🔧 Verifying ingress addon...
🌟 The 'ingress' addon is enabled
einfochips@AHMLPT1707:~/day-09$ minikube start --addons=ingress
🌟 minikube v1.33.1 on Ubuntu 22.04
🌟 Using the docker driver based on existing profile
👍 Starting "minikube" primary control-plane node in "minikube" cluster
🌐 Pulling base image v0.0.44 ...
🔧 Updating the running docker "minikube" container ...
🔧 Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
🔧 Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  ■ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
🔧 Verifying ingress addon...
```

Verify whether the ingress is running and accessible

```
einfochips@AHMLPT1707: ~/day-09
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.44 ...
Updating the running docker "minikube" container ...
Preparing Kubernetes v1.30.0 on Docker 26.1.1 ...
Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  ■ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1
  ■ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
Verifying ingress addon...
Enabled addons: storage-provisioner, default-storageclass, ingress
Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
einfochips@AHMLPT1707:~/day-09$ kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS
AGE
ingress-nginx-admission-create-wbtmq 0/1     Completed 0
15m
ingress-nginx-admission-patch-fnj86   0/1     Completed 1
15m
ingress-nginx-controller-768f948f8f-sdnf6 1/1     Running   1 (6m16s ago)
15m
einfochips@AHMLPT1707:~/day-09$
```

Create Ingress Resource:

Created frontend-deployment.yaml

```
GNU nano 6.2 frontend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: frontend
  template:
    metadata:
      labels:
        app: frontend
    spec:
      containers:
        - name: frontend
          image: nginx
          ports:
            - containerPort: 80
---
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

created backend-deployment.yaml

```
GNU nano 6.2 backend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
      - name: backend
        image: hashicorp/http-echo
        args:
        - "-text=Hello from backend"
        ports:
```

Read 33 lines

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Line

Applying the changes

```
■ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1
Verifying ingress addon...
Enabled addons: storage-provisioner, default-storageclass, ingress
Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
einfochips@AHMLPT1707:~/day-09$ kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS
AGE
ingress-nginx-admission-create-wbtmq 0/1     Completed 0
15m
ingress-nginx-admission-patch-fnj86  0/1     Completed 1
15m
ingress-nginx-controller-768f948f8f-sdnf6 1/1     Running   1 (6m16s ago)
15m
einfochips@AHMLPT1707:~/day-09$ nano frontend-deployment.yaml
einfochips@AHMLPT1707:~/day-09$ nano backend-deployment.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created
service/frontend-service created
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
service/backend-service created
einfochips@AHMLPT1707:~/day-09$ nano backend-deployment.yaml
einfochips@AHMLPT1707:~/day-09$
```

Create an ingress resource

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 ingress-resource.yaml *
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
spec:
  rules:
  - host: myapp.local
    http:
      paths:
      - path: /frontend
        pathType: Prefix
        backend:
          service:
            name: frontend-service
            port:
              number: 80
      - path: /backend
        pathType: Prefix
        backend:
          service:
            name: backend-service
            port:
              number: 80
  - path: /backend
    pathType: Prefix
    backend:
      service:
        name: backend-service
        port:
          number: 80

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Apply the ingress resource

```
einfochips@AHMLPT1707: ~/day-09
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" name
space by default
einfochips@AHMLPT1707:~/day-09$ kubectl get pods -n ingress-nginx
NAME                                READY   STATUS    RESTARTS
AGE
ingress-nginx-admission-create-wbtmq 0/1     Completed 0
15m
ingress-nginx-admission-patch-fnj86   0/1     Completed 1
15m
ingress-nginx-controller-768f948f8f-sdnf6 1/1     Running   1 (6m16s ago)
15m
einfochips@AHMLPT1707:~/day-09$ nano frontend-deployment.yaml
einfochips@AHMLPT1707:~/day-09$ nano backend-deployment.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created
service/frontend-service created
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
service/backend-service created
einfochips@AHMLPT1707:~/day-09$ nano backend-deployment.yaml
einfochips@AHMLPT1707:~/day-09$ nano ingress-resource.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f ingress-resource.yaml
ingress.networking.k8s.io/example-ingress created
einfochips@AHMLPT1707:~/day-09$
```


Configure URL rewriting in the ingress resource to modify incoming URLs before they reach the backend services.

Enable sticky sessions to ensure that requests from the same client are directed to the same backend pod.

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 ingress-resource.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  annotations:
    nginx.ingress.kubernetes.io/affinity: "cookie"
    nginx.ingress.kubernetes.io/session-cookie-name: "route"
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
  - host: myapp.local
    http:
      paths:
      - path: /frontend
        pathType: Prefix
        backend:
          service:
            name: frontend-service
            port:
              number: 80
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 ingress-resource.yaml
      name: frontend-service
      port:
        number: 80
    - path: /backend
      pathType: Prefix
      backend:
        service:
          name: backend-service
          port:
            number: 80
  - host: myapp.local
    http:
      paths:
      - path: /oldpath/(.*)
        pathType: Prefix
        backend:
          service:
            name: new-service
            port:
              number: 80
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line

3: Implementing Horizontal Pod Autoscaling

Configure Horizontal Pod Autoscaler:

Write a horizontal pod autoscaler (HPA) manifest to automatically scale the static web application pods based on CPU utilization.

```
einfochips@AHMLPT1707: ~/day-09
GNU nano 6.2 hpa.yaml *
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: mywebapp-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: mywebapp-deployment
  minReplicas: 2
  maxReplicas: 5
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Applying hpa.yaml

```
einfochips@AHMLPT1707: ~/day-09
einfochips@AHMLPT1707:~/day-09$ minikube ip
192.168.49.2
einfochips@AHMLPT1707:~/day-09$ sudo nano /etc/hosts
einfochips@AHMLPT1707:~/day-09$ curl http://myapp.local/frontend
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body>
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>nginx</center>
</body>
</html>
einfochips@AHMLPT1707:~/day-09$ curl http://myapp.local/frontend
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body>
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>nginx</center>
</body>
</html>
einfochips@AHMLPT1707:~/day-09$ nano ingress-resource.yaml
einfochips@AHMLPT1707:~/day-09$ nano hpa.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/mywebapp-hpa created
einfochips@AHMLPT1707:~/day-09$
```

Stres testing

Perform stress testing to simulate traffic and validate the HPA configuration.

Monitor the scaling behavior and ensure the application scales up and down based on the load.

```
einfochips@AHMLPT1707: ~/day-09
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>nginx</center>
</body>
</html>
einfochips@AHMLPT1707:~/day-09$ curl http://myapp.local/frontend
<html>
<head><title>503 Service Temporarily Unavailable</title></head>
<body>
<center><h1>503 Service Temporarily Unavailable</h1></center>
<hr><center>nginx</center>
</body>
</html>
einfochips@AHMLPT1707:~/day-09$ nano ingress-resource.yaml
einfochips@AHMLPT1707:~/day-09$ nano hpa.yaml
einfochips@AHMLPT1707:~/day-09$ kubectl apply -f hpa.yaml
horizontalpodautoscaler.autoscaling/mywebapp-hpa created
einfochips@AHMLPT1707:~/day-09$ kubectl get hpa
```

NAME	REFERENCE	TARGETS	MINPODS
MAXPODS	REPLICAS	AGE	
mywebapp-hpa	Deployment/mywebapp-deployment	cpu: <unknown>/50%	2
5	0	2m10s	
nodejs-app-hpa	Deployment/nodejs-app-deployment	cpu: <unknown>/50%	2
5	2	27h	

```
einfochips@AHMLPT1707:~/day-09$
```

```
NAME          REFERENCE          TARGETS          MINPODS  MAXPODS  REPLICAS  AGE
mywebapp-hpa  Deployment/backend  cpu: <unknown>/2%  1         10        2         22h
mywebapp-hpa  Deployment/backend  cpu: 10%/2%       1         10        2         22h
mywebapp-hpa  Deployment/backend  cpu: 10%/2%       1         10        4         22h
mywebapp-hpa  Deployment/backend  cpu: 10%/2%       1         10        5         22h
mywebapp-hpa  Deployment/backend  cpu: 315%/2%      1         10        5         22h
mywebapp-hpa  Deployment/backend  cpu: 315%/2%      1         10        10        22h
mywebapp-hpa  Deployment/backend  cpu: 18%/2%       1         10        10        22h
```

```
error: the server doesn't have a resource type "backend-deployment"
einfochips@AHMLPT1707:~/day-09$ kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
backend	10/10	10	10	23h


```
einfochips@AHMLPT1707: ~/day-09
To https://github.com/yashmahi88/day09.git
* [new branch]      master -> master
einfochips@AHMLPT1707:~/day-09$ curl http://myapp.local/
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
einfochips@AHMLPT1707:~/day-09$ curl http://myapp.local/api/
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
</body>
</html>
einfochips@AHMLPT1707:~/day-09$ curl http://myapp.local/api/v2
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx</center>
```

Adding TLS Secret to the ingress-resource.yaml

```
GNU nano 6.2 ingress-resource.yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  annotations:
    nginx.ingress.kubernetes.io/affinity: "cookie"
    nginx.ingress.kubernetes.io/session-cookie-name: "route"
    nginx.ingress.kubernetes.io/rewrite-target: /
    nginx.ingress.kubernetes.io/ssl-redirect: "true"
spec:
  tls:
  - hosts:
    - myapp.local
    secretName: my-tls-secret
  rules:
  - host: myapp.local
    http:
      paths:
      - path: /frontend
        pathType: Prefix

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

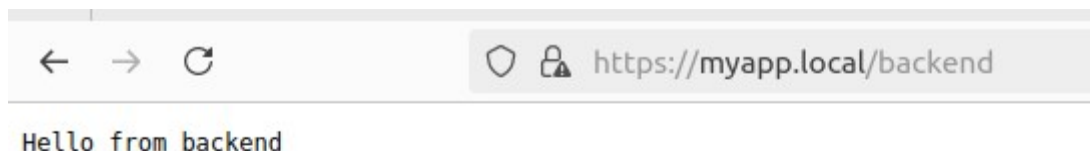
Creating SSL certificate

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj  
"/CN=myapp.ingress/O=ingressapp-tls"
```

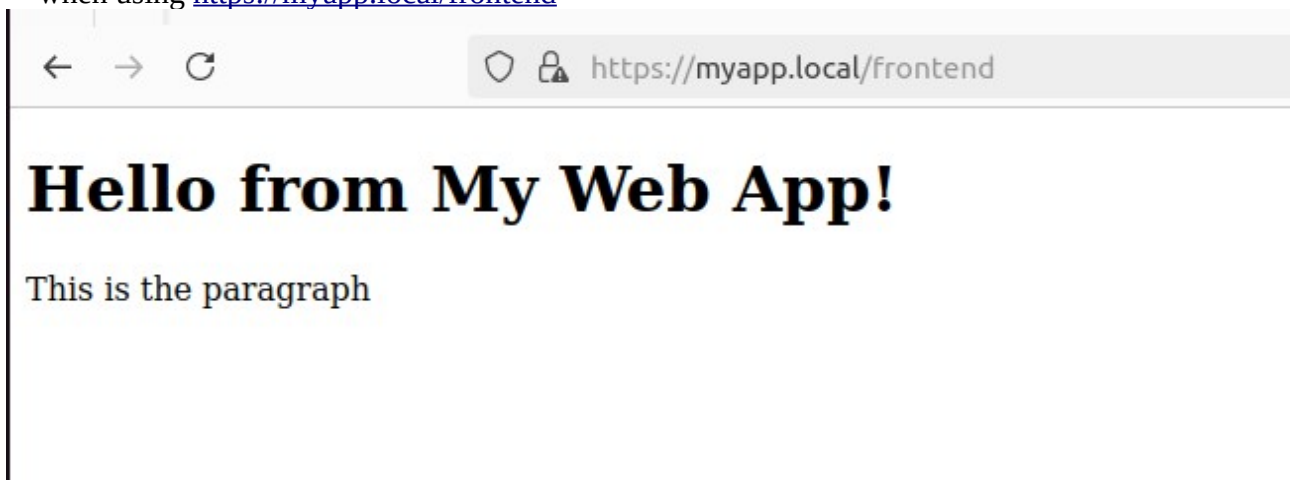
```
kubectl create secret tls my-tls-secret --cert=tls.crt --key=tls.key
```

```
einfochips@AHMLPT1707:~/day-09$ kubectl create secret tls my-tls-secret --cert=tls.crt --key=tls.key  
secret/my-tls-secret created  
einfochips@AHMLPT1707:~/day-09$ nano ingress-resource.yaml  
einfochips@AHMLPT1707:~/day-09$ curl https://myapp.local/frontend  
curl: (60) SSL certificate problem: self-signed certificate  
More details here: https://curl.se/docs/sslcerts.html  
  
curl failed to verify the legitimacy of the server and therefore could not  
establish a secure connection to it. To learn more about this situation and  
how to fix it, please visit the web page mentioned above.  
einfochips@AHMLPT1707:~/day-09$ curl https://myapp.local/backend  
curl: (60) SSL certificate problem: self-signed certificate  
More details here: https://curl.se/docs/sslcerts.html  
  
curl failed to verify the legitimacy of the server and therefore could not  
establish a secure connection to it. To learn more about this situation and  
how to fix it, please visit the web page mentioned above.  
einfochips@AHMLPT1707:~/day-09$
```

When using <https://myapp.local/backend>



when using <https://myapp.local/frontend>



NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
mywebapp-hpa	Deployment/backend	cpu: <unknown>/2%	1	10	2	22h
mywebapp-hpa	Deployment/backend	cpu: 10%/2%	1	10	2	22h
mywebapp-hpa	Deployment/backend	cpu: 10%/2%	1	10	4	22h
mywebapp-hpa	Deployment/backend	cpu: 10%/2%	1	10	5	22h
mywebapp-hpa	Deployment/backend	cpu: 315%/2%	1	10	5	22h
mywebapp-hpa	Deployment/backend	cpu: 315%/2%	1	10	10	22h
mywebapp-hpa	Deployment/backend	cpu: 18%/2%	1	10	10	22h

error: the server doesn't have a resource type "backend-deployment"

• **einfochips@AHMLPT1707:~/day-09\$** kubectl get deployments

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
backend	10/10	10	10	23h