
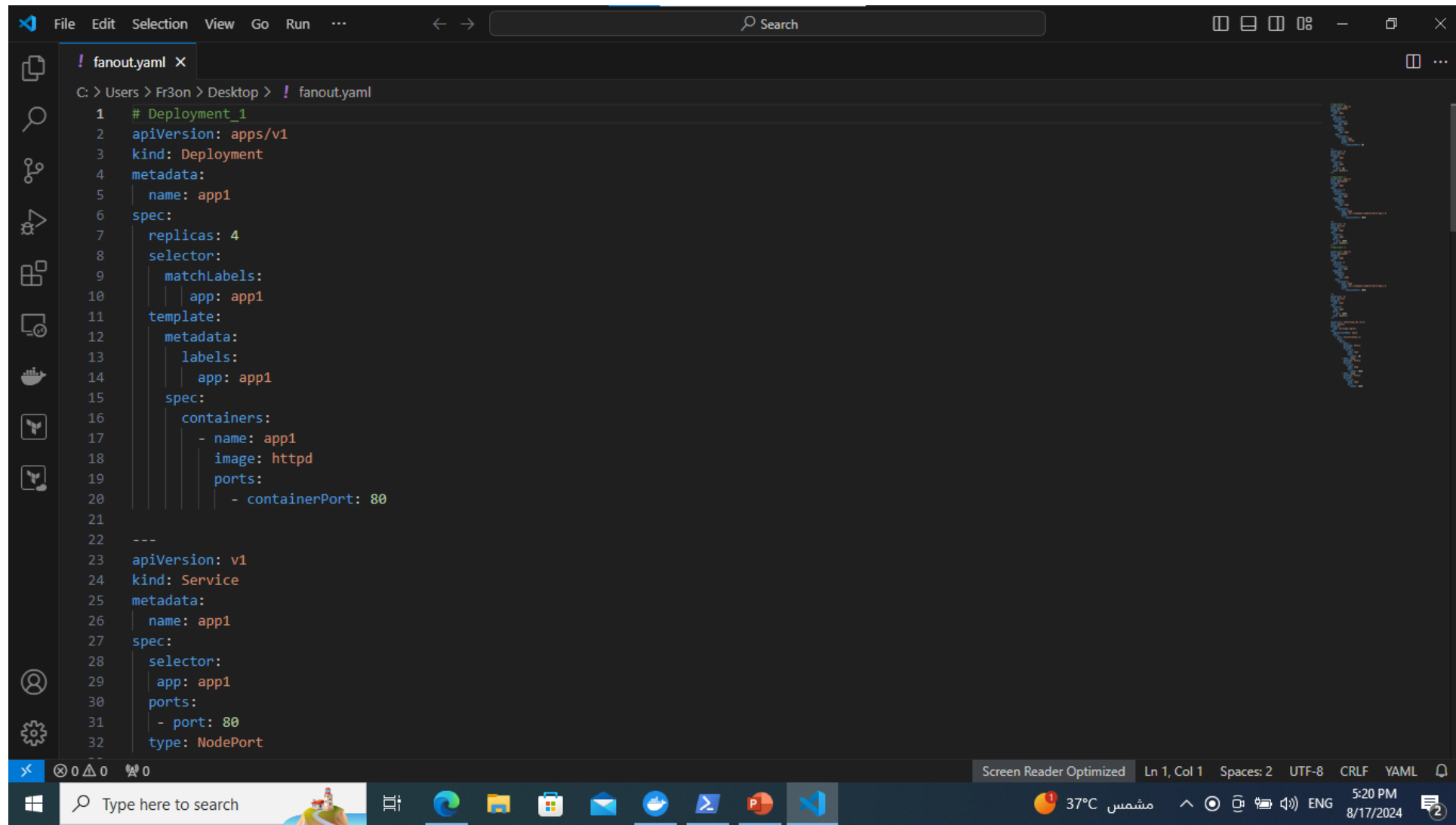


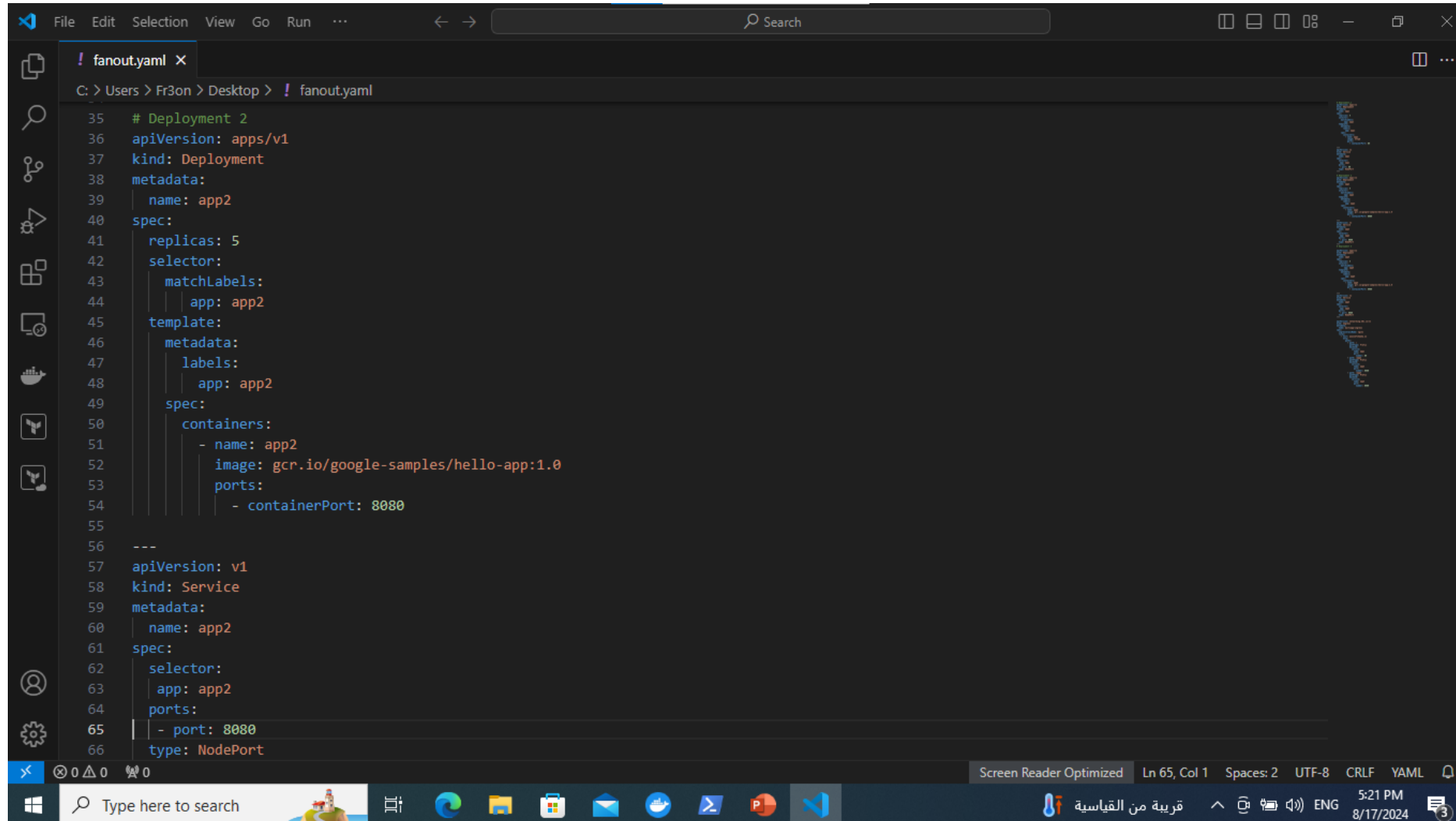
- 
- In this project,
 - Deployed 3 applications: an httpd Apache web server and two instances of Google Hello World (versions 1 and 2).
 - Exposed each application with its respective service.
 - Configured an Ingress resource to define hostname-based routing, paths, and service mappings.
 - Enabled the Ingress controller addon in Minikube.
 - Accessed Minikube via SSH and mapped the hostname to the Minikube host IP to enable seamless access.

Setting-up the first deployment and exposing it



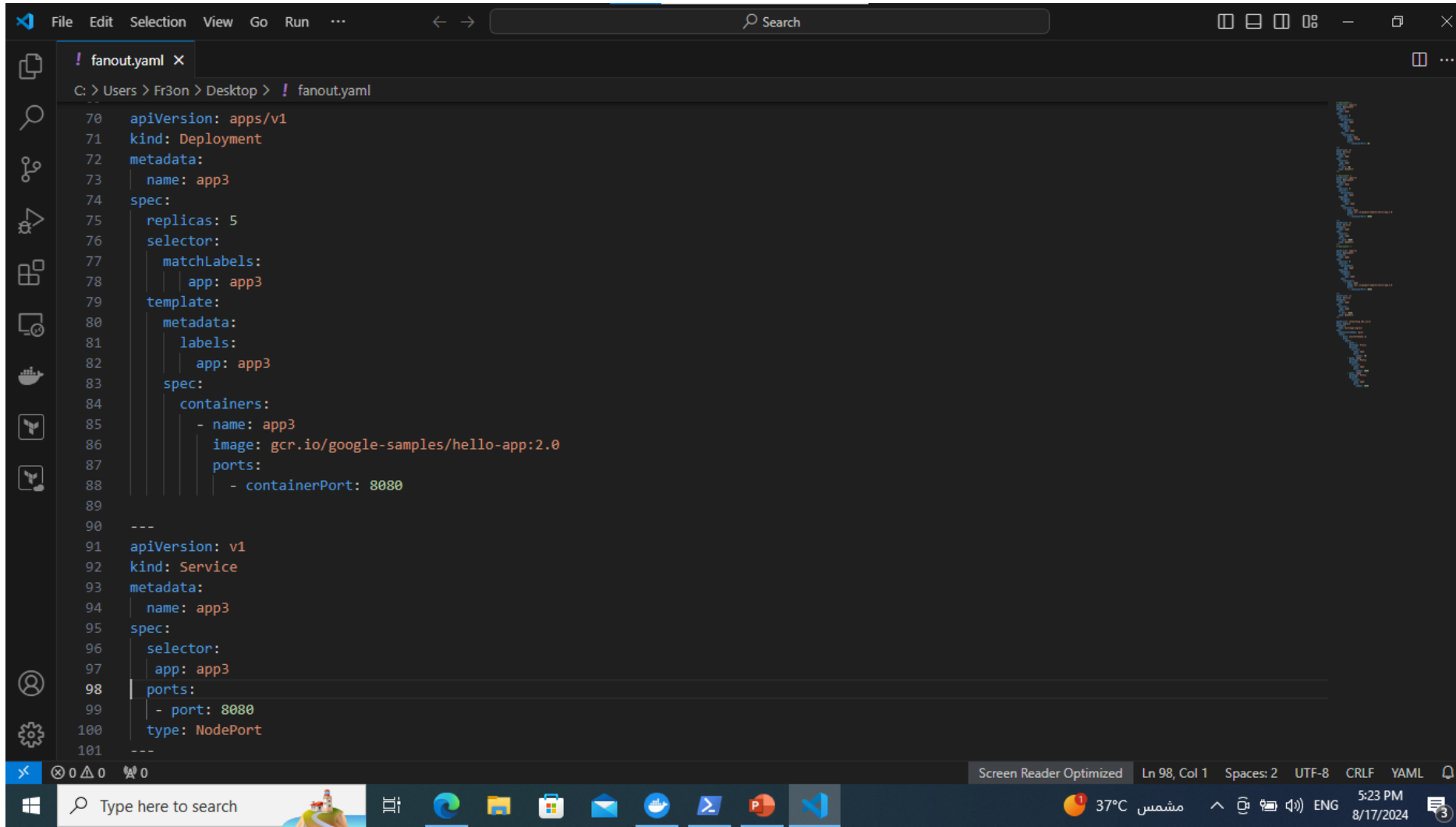
```
1 # Deployment_1
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: app1
6 spec:
7   replicas: 4
8   selector:
9     matchLabels:
10      app: app1
11   template:
12     metadata:
13       labels:
14         app: app1
15     spec:
16       containers:
17       - name: app1
18         image: httpd
19         ports:
20         - containerPort: 80
21
22 ---
23 apiVersion: v1
24 kind: Service
25 metadata:
26   name: app1
27 spec:
28   selector:
29     app: app1
30   ports:
31   - port: 80
32     type: NodePort
```

Second deployment with exposing its service



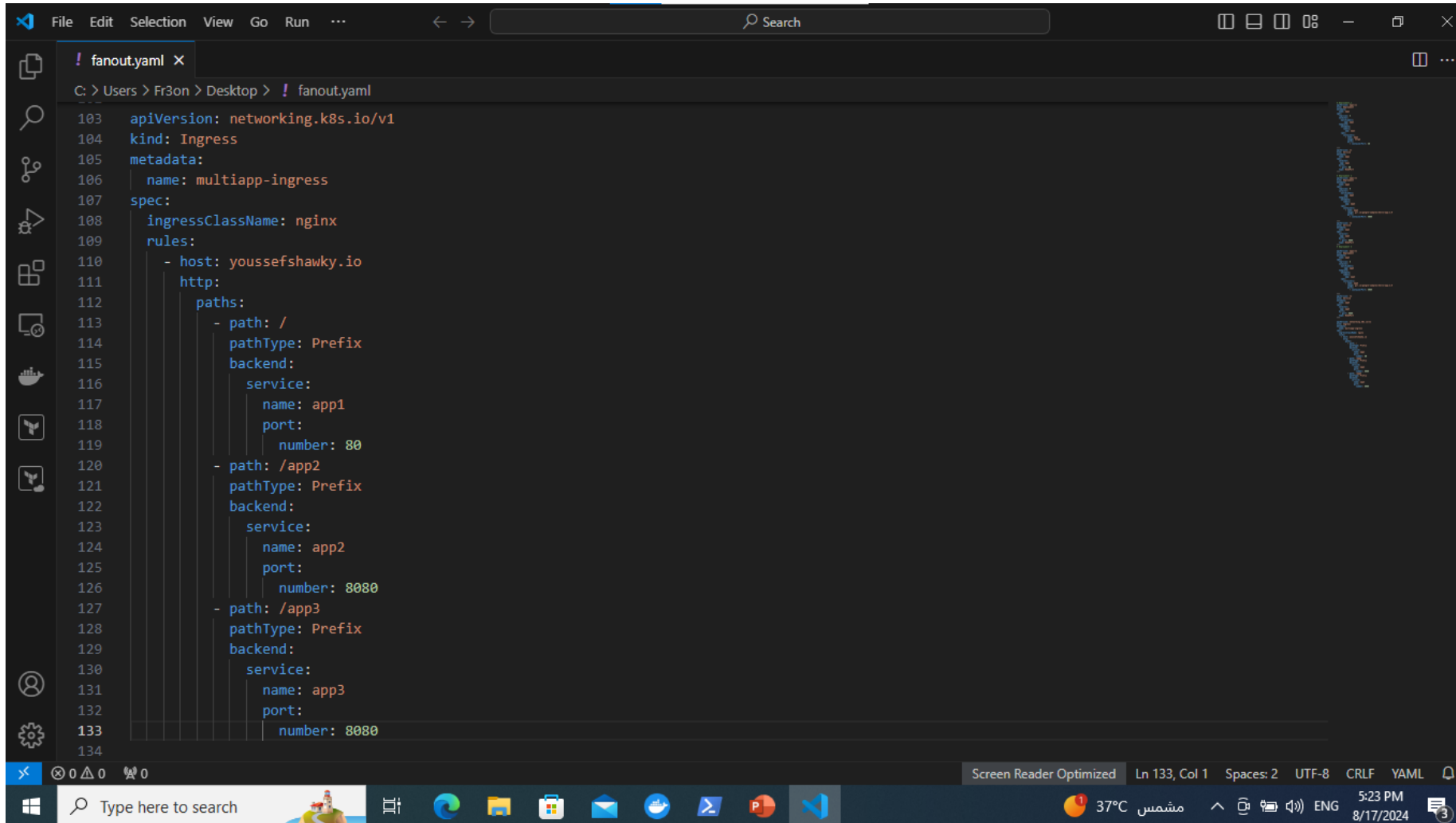
```
35 # Deployment 2
36 apiVersion: apps/v1
37 kind: Deployment
38 metadata:
39   name: app2
40 spec:
41   replicas: 5
42   selector:
43     matchLabels:
44       app: app2
45   template:
46     metadata:
47       labels:
48         app: app2
49     spec:
50       containers:
51       - name: app2
52         image: gcr.io/google-samples/hello-app:1.0
53         ports:
54         - containerPort: 8080
55 ---
56
57 apiVersion: v1
58 kind: Service
59 metadata:
60   name: app2
61 spec:
62   selector:
63     app: app2
64   ports:
65   - port: 8080
66     type: NodePort
```

Third deployment



```
70 apiVersion: apps/v1
71 kind: Deployment
72 metadata:
73   name: app3
74 spec:
75   replicas: 5
76   selector:
77     matchLabels:
78       app: app3
79   template:
80     metadata:
81       labels:
82         app: app3
83     spec:
84       containers:
85       - name: app3
86         image: gcr.io/google-samples/hello-app:2.0
87         ports:
88         - containerPort: 8080
89
90 ---
91 apiVersion: v1
92 kind: Service
93 metadata:
94   name: app3
95 spec:
96   selector:
97     app: app3
98   ports:
99   - port: 8080
100     type: NodePort
101 ---
```

Setting-up the ingress recourse

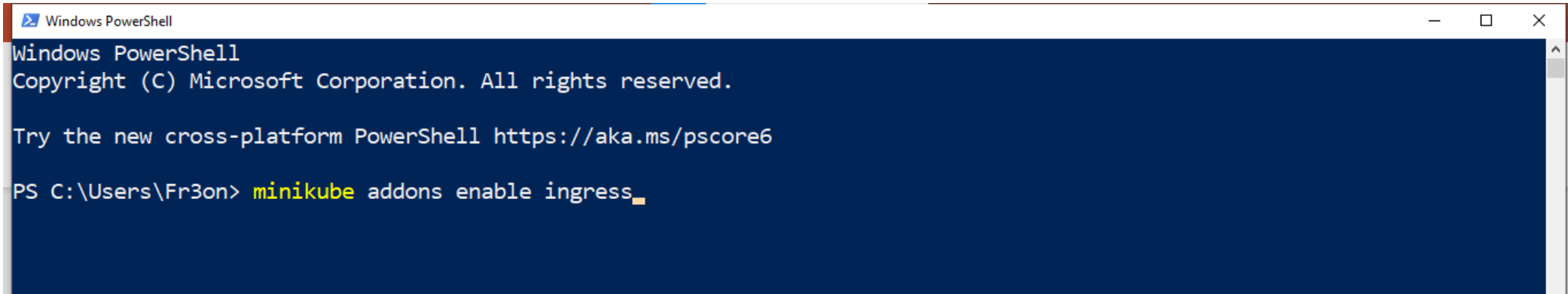


```
103 apiVersion: networking.k8s.io/v1
104 kind: Ingress
105 metadata:
106   name: multiapp-ingress
107 spec:
108   ingressClassName: nginx
109   rules:
110     - host: youssefshawky.io
111       http:
112         paths:
113           - path: /
114             pathType: Prefix
115             backend:
116               service:
117                 name: app1
118                 port:
119                   number: 80
120           - path: /app2
121             pathType: Prefix
122             backend:
123               service:
124                 name: app2
125                 port:
126                   number: 8080
127           - path: /app3
128             pathType: Prefix
129             backend:
130               service:
131                 name: app3
132                 port:
133                   number: 8080
134
```

Applying the file to run

```
PS C:\Users\Fr3on> kubectl apply -f fanout.yaml
```

By default the ingress-controller is not coming with the minikube, you need to enable it manually

A screenshot of a Windows PowerShell terminal window. The window has a title bar that says "Windows PowerShell" and standard Windows window controls (minimize, maximize, close). The terminal background is dark blue with white text. The text displayed is: "Windows PowerShell", "Copyright (C) Microsoft Corporation. All rights reserved.", "Try the new cross-platform PowerShell https://aka.ms/pscore6", and "PS C:\Users\Fr3on> minikube addons enable ingress_".

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Fr3on> minikube addons enable ingress_
```

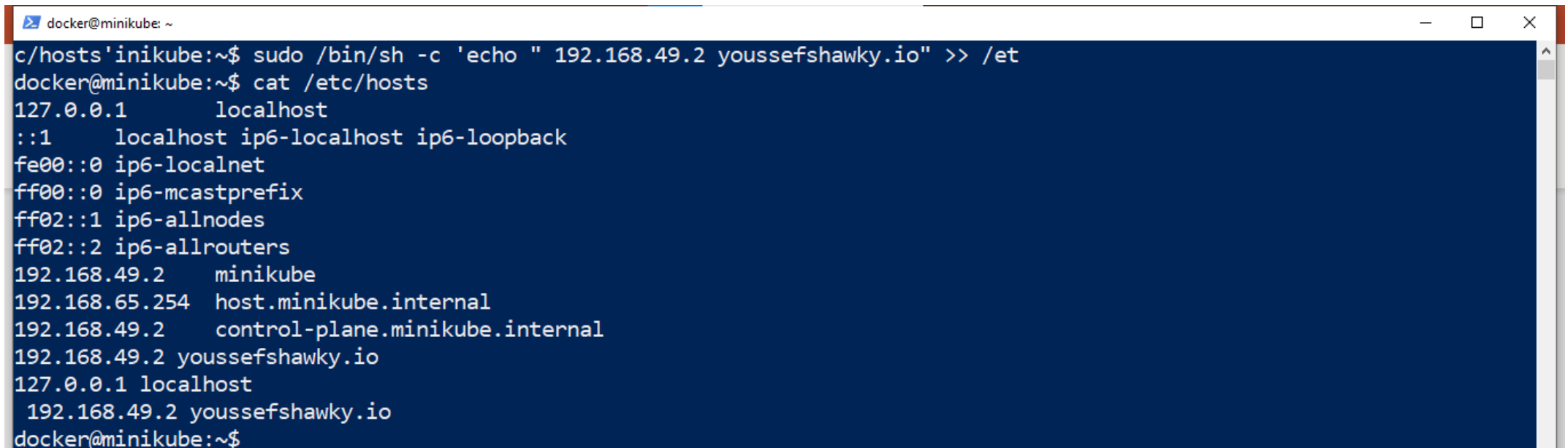
Checking host is mapped to address, usually be minikube address, if not the same you must map it

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Fr3on> kubectl get ingress
NAME          CLASS  HOSTS          ADDRESS      PORTS    AGE
multiapp-ingress  nginx  youssefshawky.io  192.168.49.2  80      171m
PS C:\Users\Fr3on> minikube ip
192.168.49.2
PS C:\Users\Fr3on>
```


Accessing minikube via ssh then map the host to minikube host page



```
docker@minikube: ~  
c:/hosts'inikube:~$ sudo /bin/sh -c 'echo " 192.168.49.2 youssefshawky.io" >> /et  
docker@minikube:~$ cat /etc/hosts  
127.0.0.1      localhost  
::1           localhost ip6-localhost ip6-loopback  
fe00::0       ip6-localnet  
ff00::0       ip6-mcastprefix  
ff02::1       ip6-allnodes  
ff02::2       ip6-allrouters  
192.168.49.2   minikube  
192.168.65.254 host.minikube.internal  
192.168.49.2   control-plane.minikube.internal  
192.168.49.2   youssefshawky.io  
127.0.0.1      localhost  
192.168.49.2   youssefshawky.io  
docker@minikube:~$
```

Accessing the first service

A terminal window with a dark blue background and white text. The window title bar shows 'docker@minikube: ~' and standard window controls. The terminal shows a user running a curl command to access a service, which returns an HTML response.

```
docker@minikube:~$ curl youssefshawky.io
<html><body><h1>It works!</h1></body></html>
docker@minikube:~$
```

Second service

A terminal window with a dark blue background and white text. The window title bar shows 'docker@minikube: ~' and standard window controls. The terminal content shows a user running a curl command to a specific URL, receiving a 'Hello, world!' response, and then displaying version and hostname information.

```
docker@minikube: ~  
docker@minikube:~$ curl youssefshawky.io/app2  
Hello, world!  
Version: 1.0.0  
Hostname: app2-7fcc489769-xnh2q  
docker@minikube:~$
```

Third service, Same hello world but different version

A terminal window with a dark blue background and white text. The window title bar shows 'docker@minikube: ~' and standard window controls. The terminal content shows a user running a curl command to a specific endpoint, receiving a 'Hello, world!' message and version information.

```
docker@minikube: ~  
docker@minikube:~$ curl youssefshawky.io/app3  
Hello, world!  
Version: 2.0.0  
Hostname: app3-74f699c794-p9hj7  
docker@minikube:~$
```



Thank you