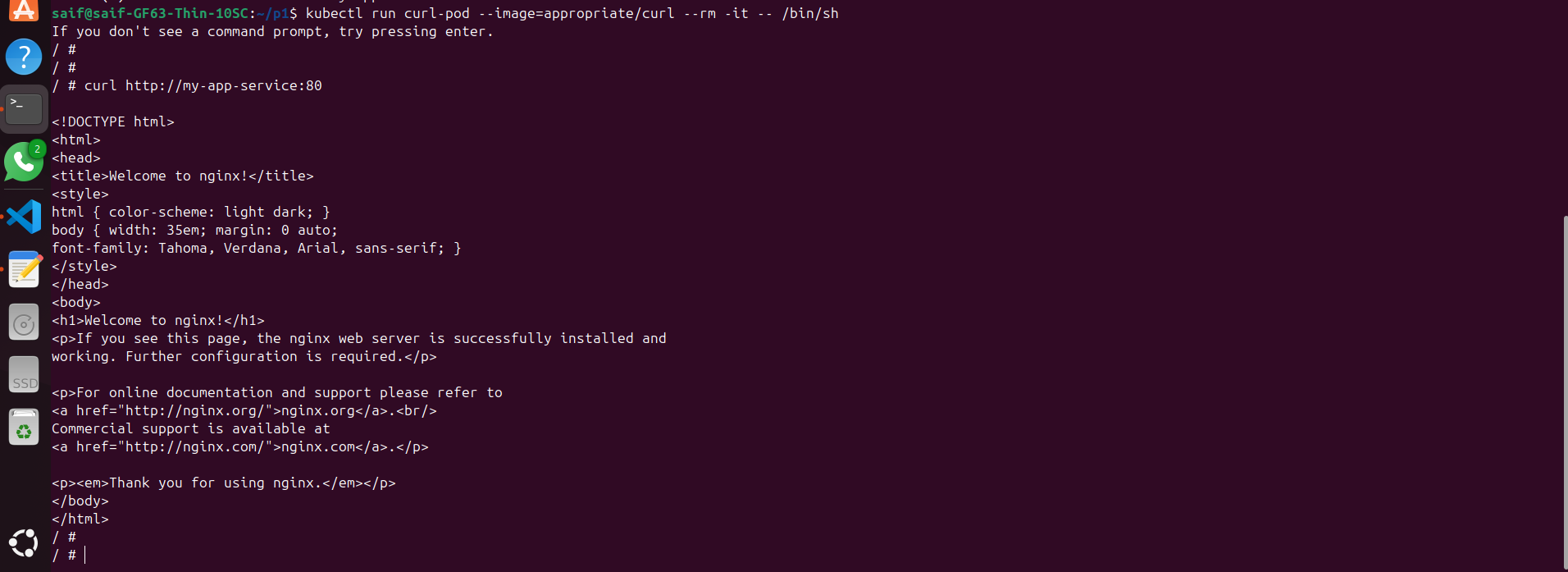
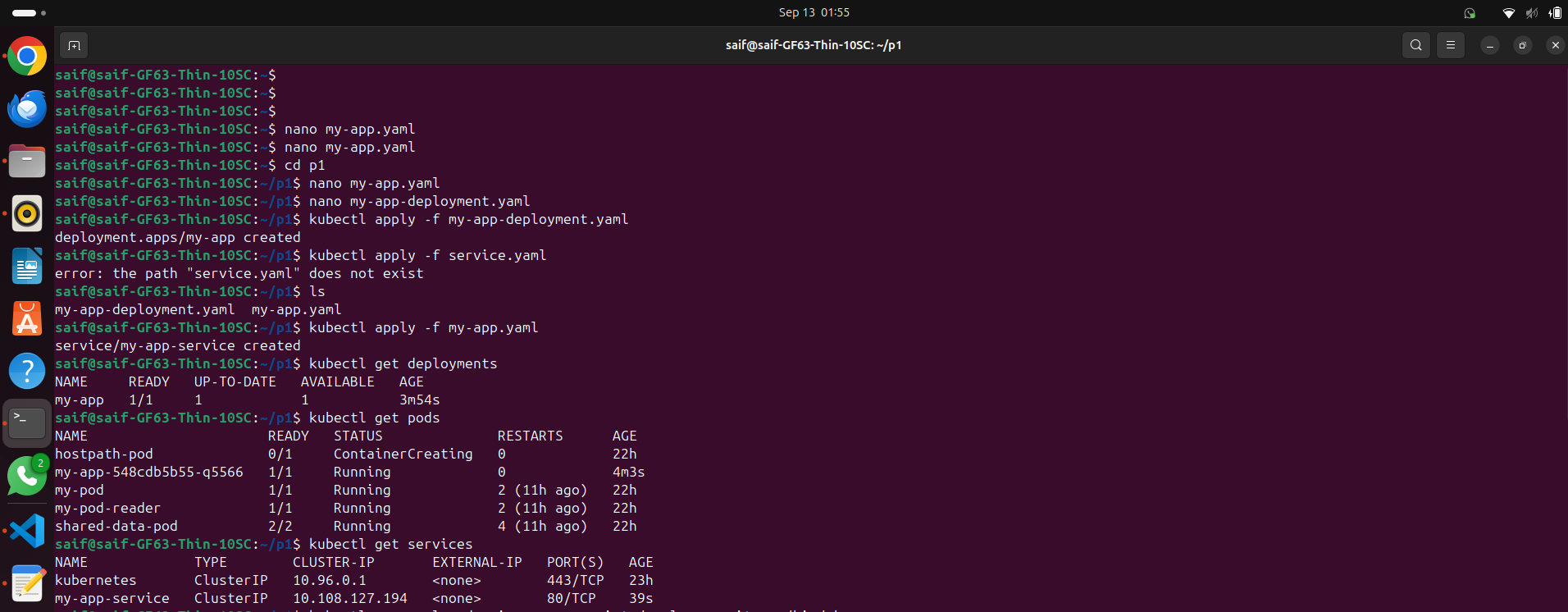
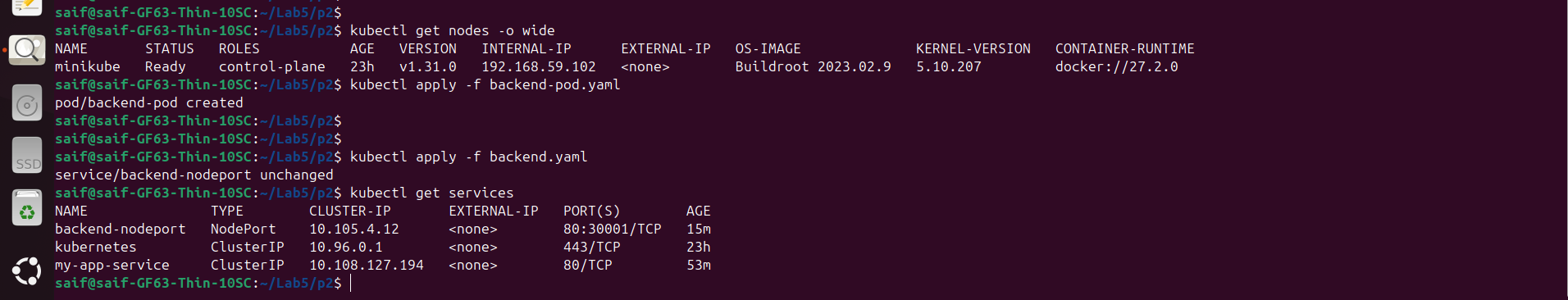
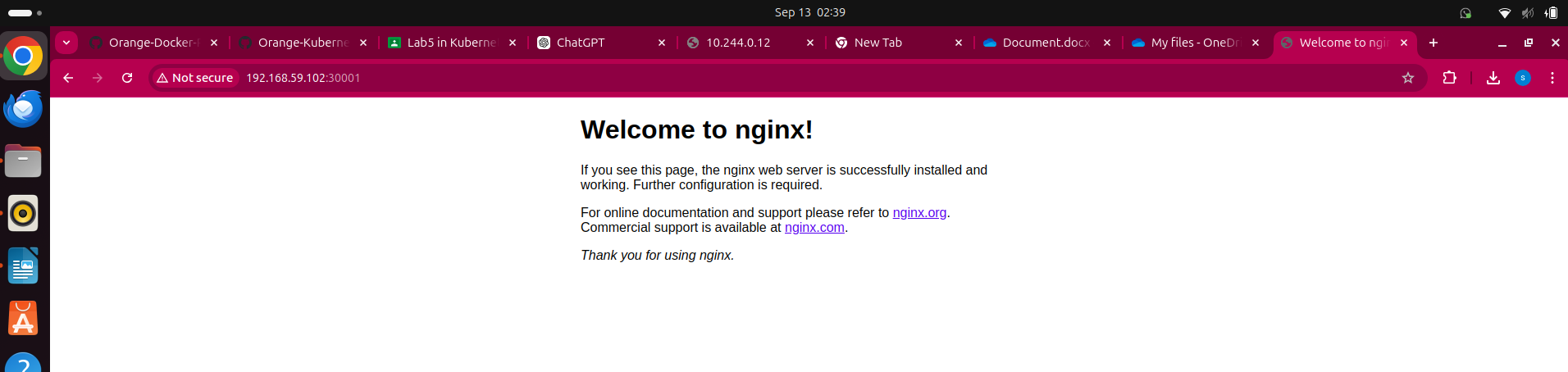
- Create a ClusterIP Service:  
 Write a YAML definition for a ClusterIP service that exposes a deployment named my-app running on port 8080. Ensure the service maps to port 80 inside the cluster.



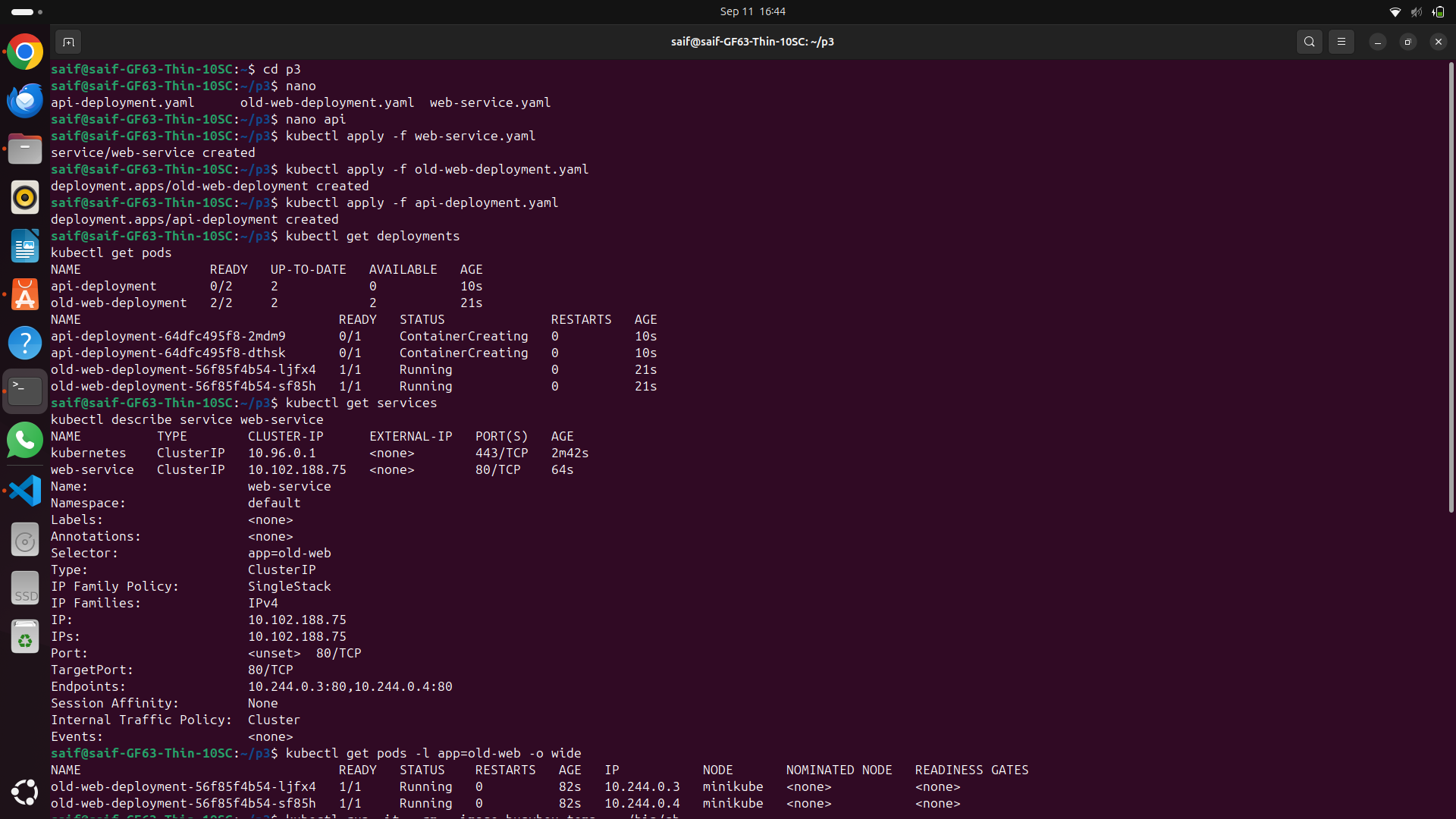
- Set up a NodePort Service:

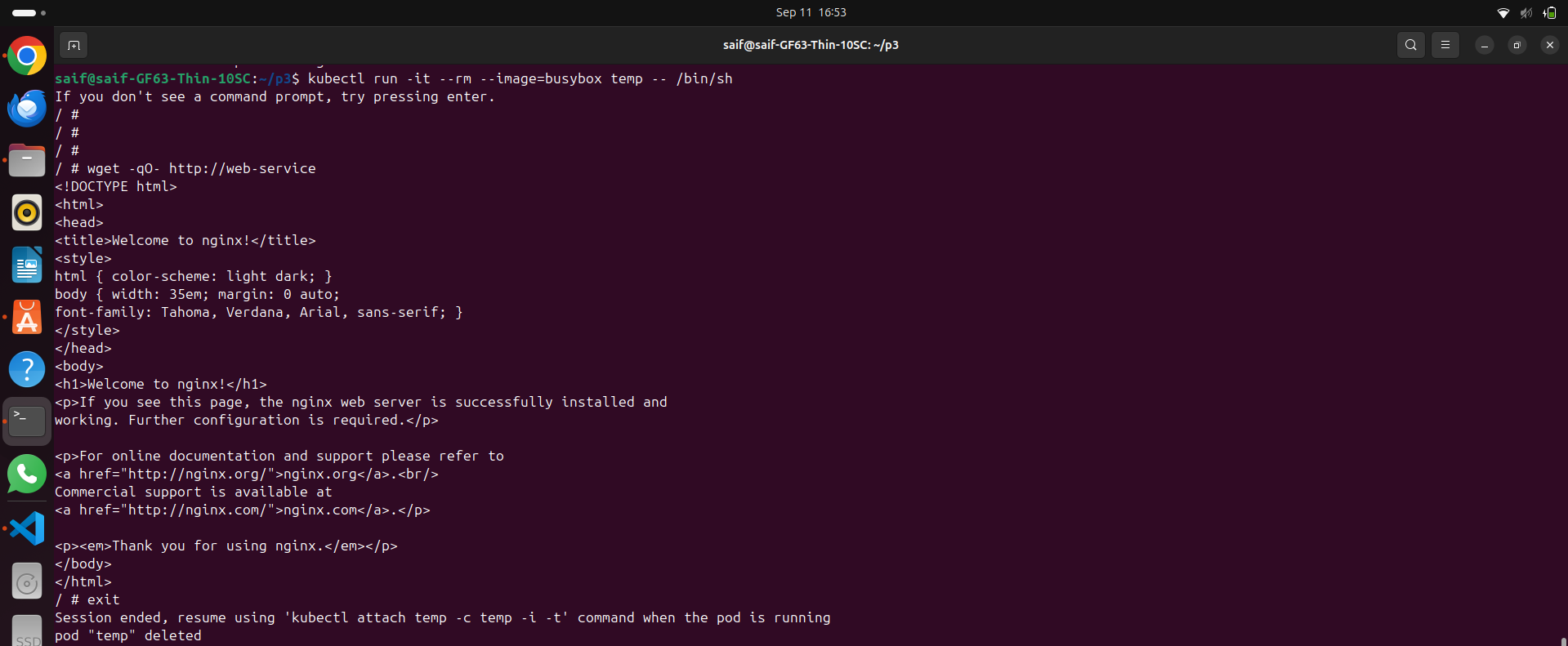
Create a NodePort service that exposes a deployment named backend on port 30001 of each node. Verify that the service is accessible externally via the node’s IP and port 30001.

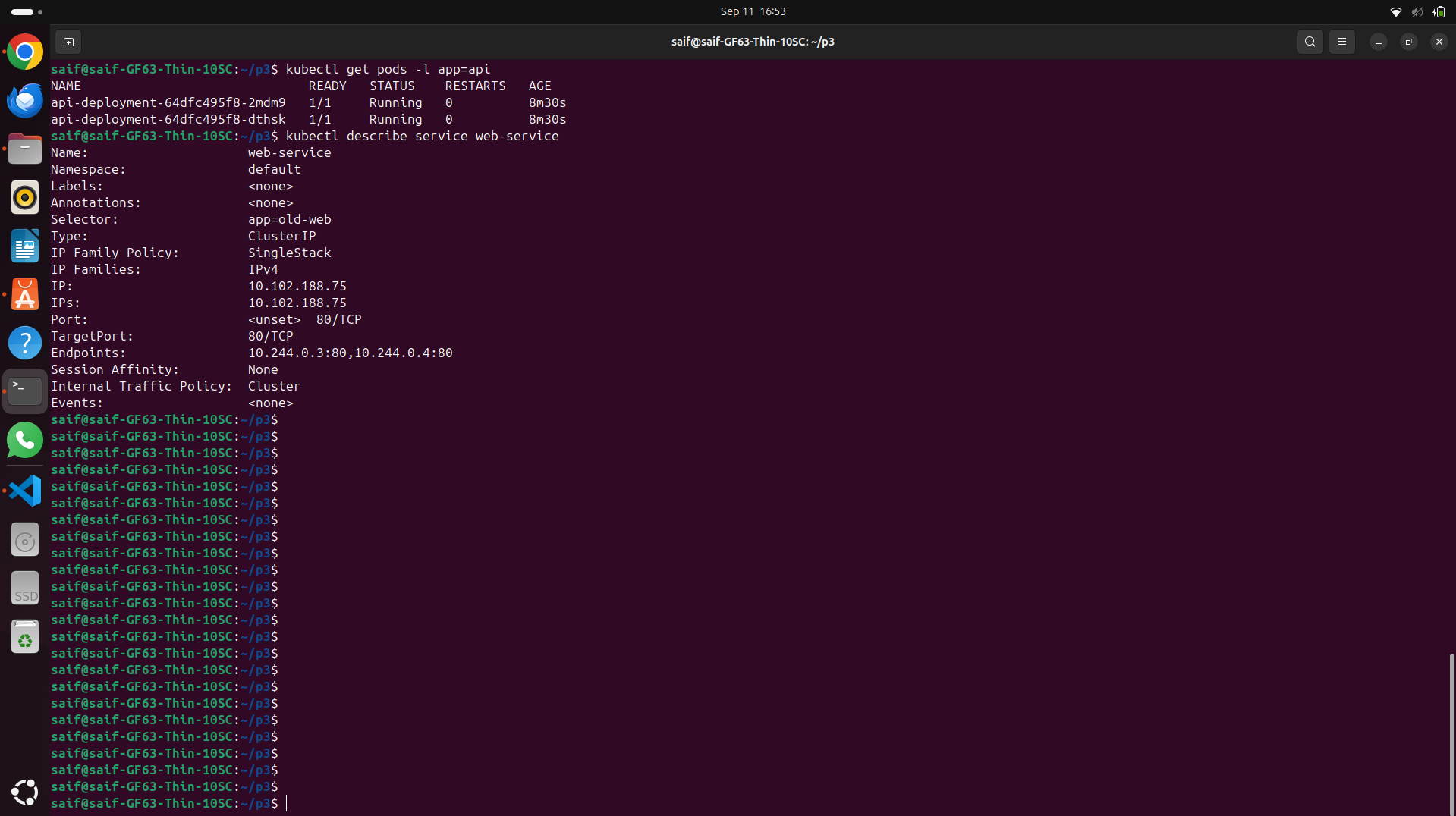




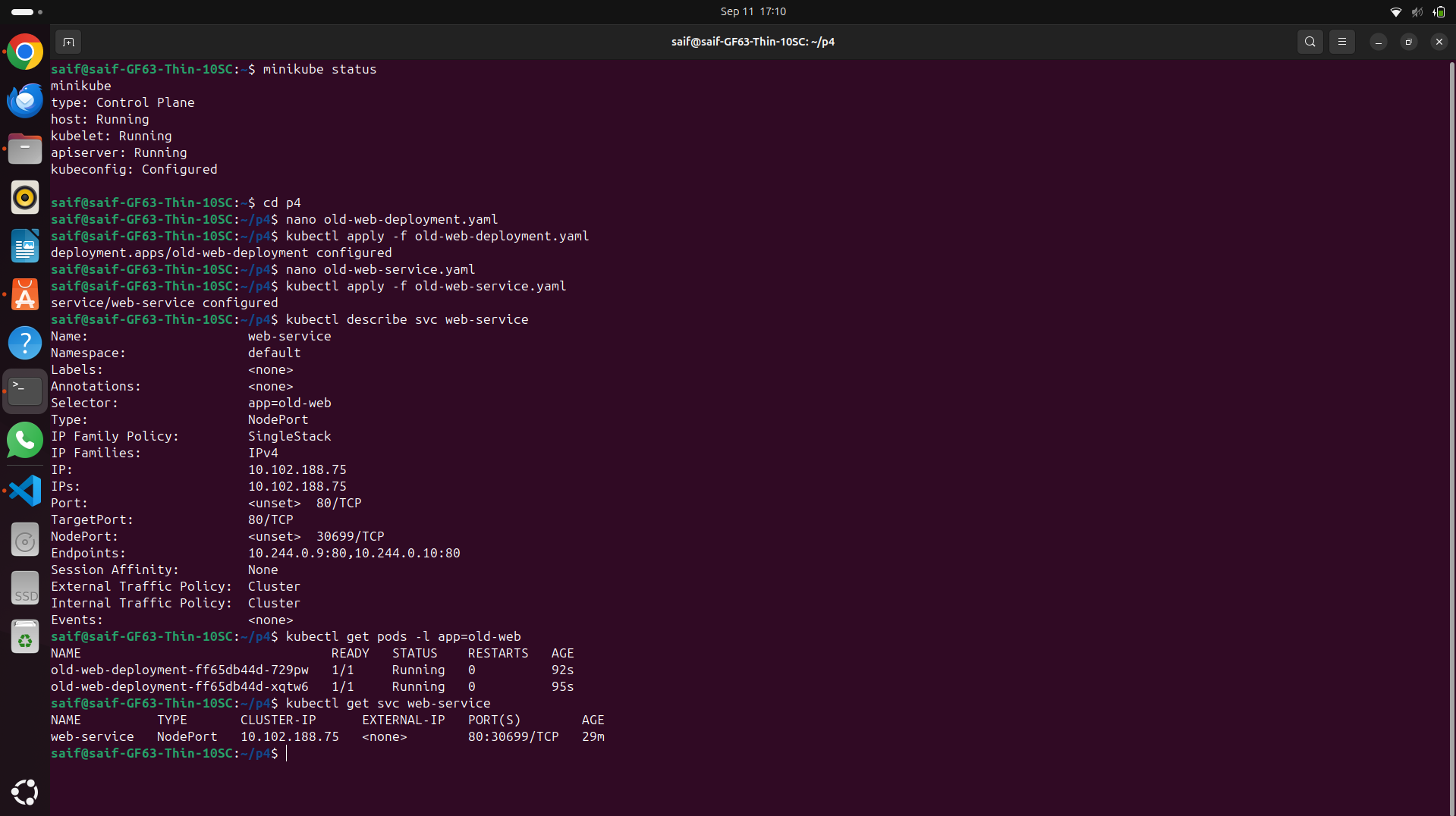
- Test Service with Selectors:  
 Create a ClusterIP service named web-service with the selector app: 0ld-web. Deploy two sets of Pods, one with the label app: old-web and another with the label app: api. Verify that only the web Pods receive traffic.

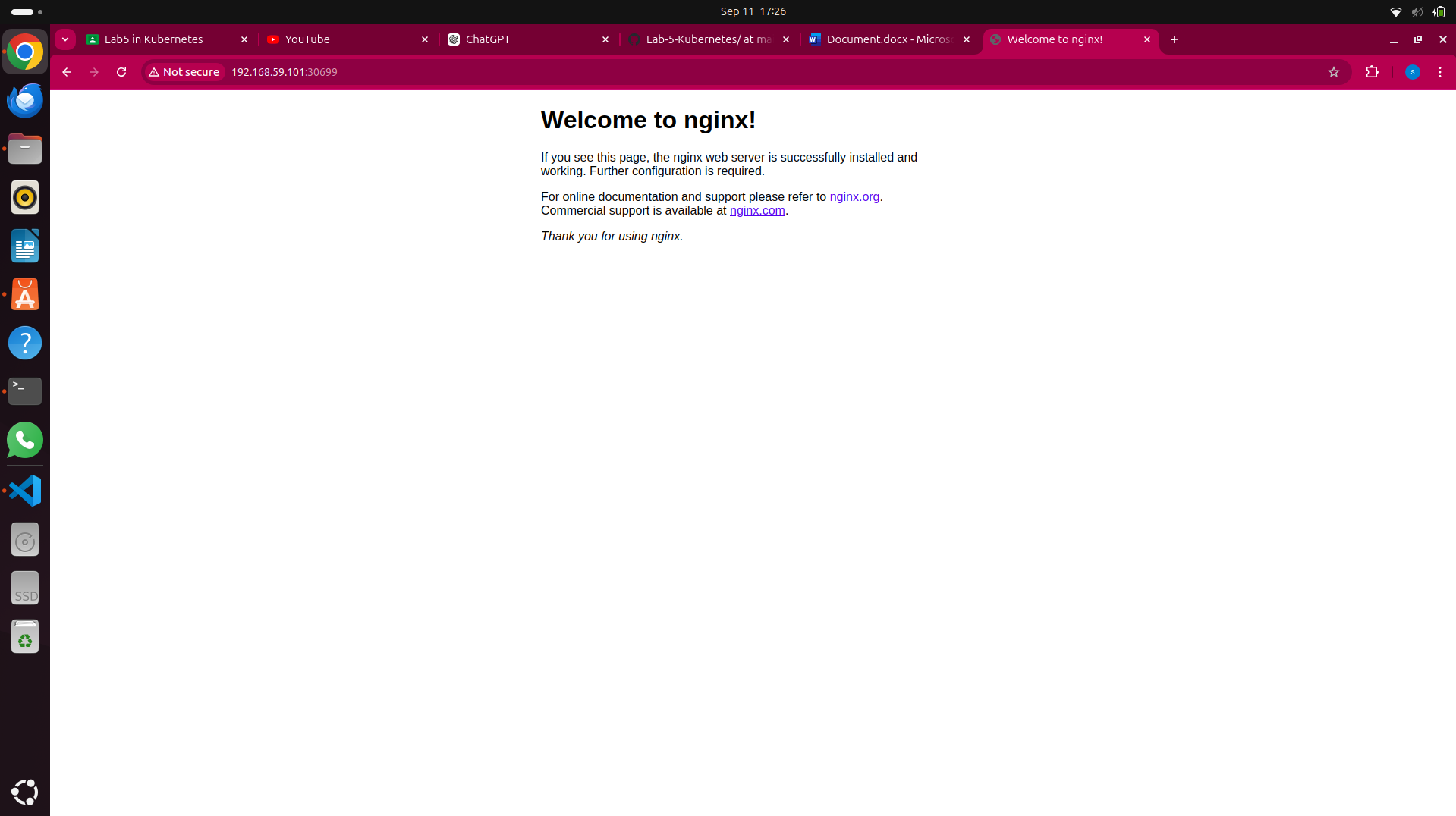


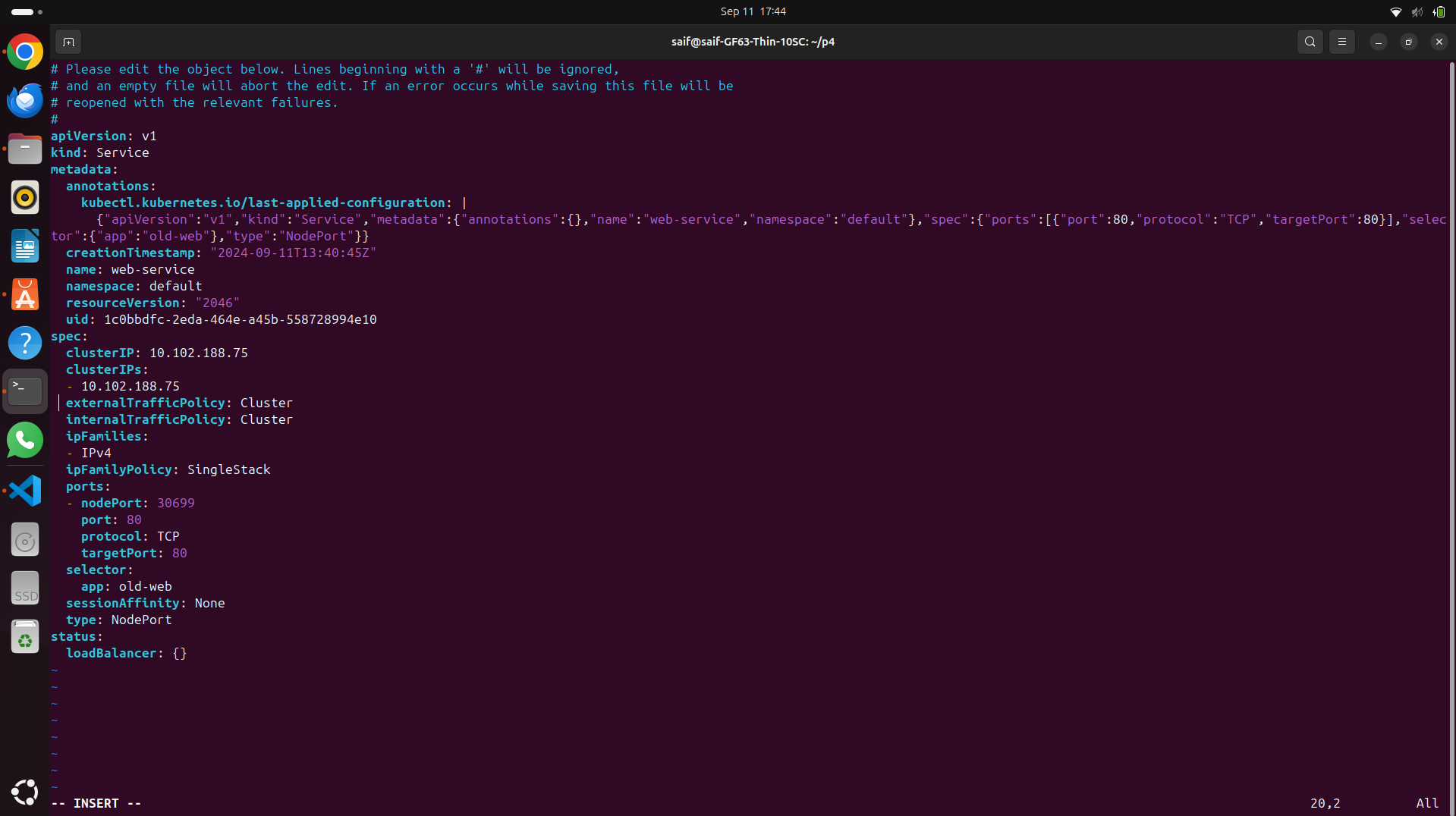


since the api Pods are listed in this section, they are receiving traffic from the web-service.

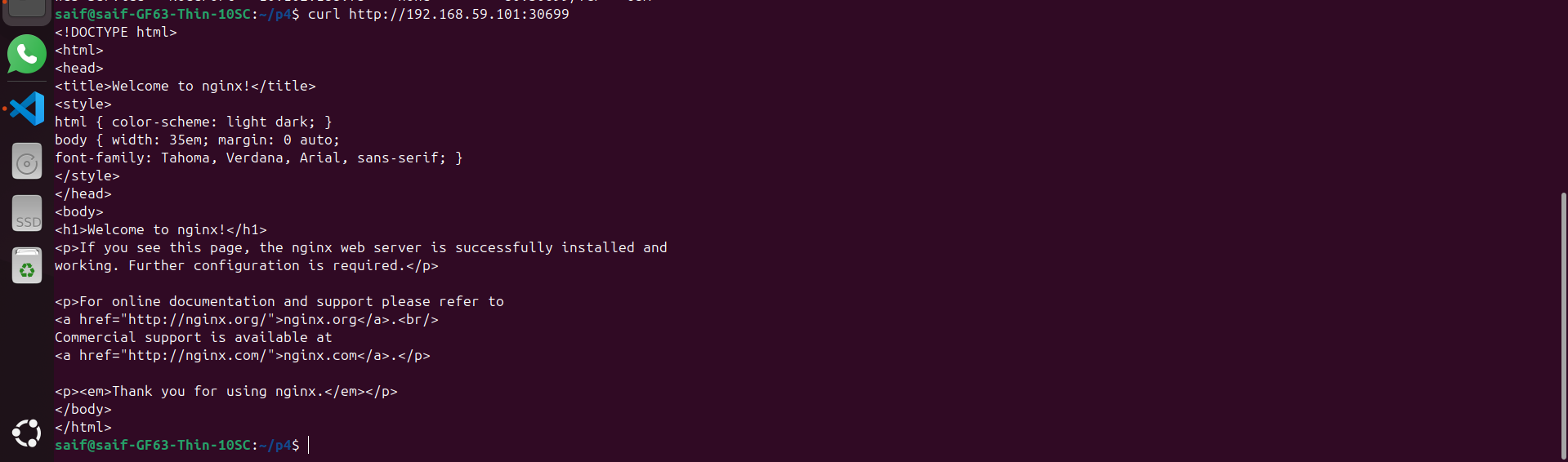
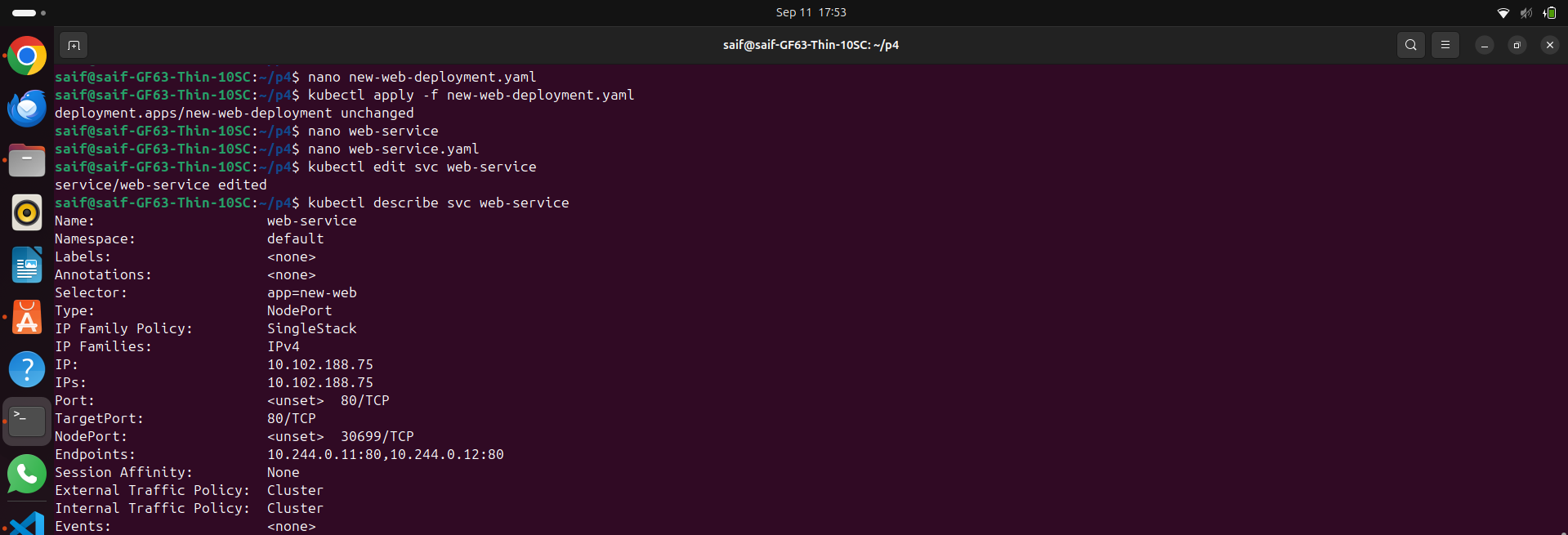
- Change Service Selector:  
 Update an existing service to change its selector from app: old-web to app: new-web. Verify that the traffic is now routed only to Pods with the new label.



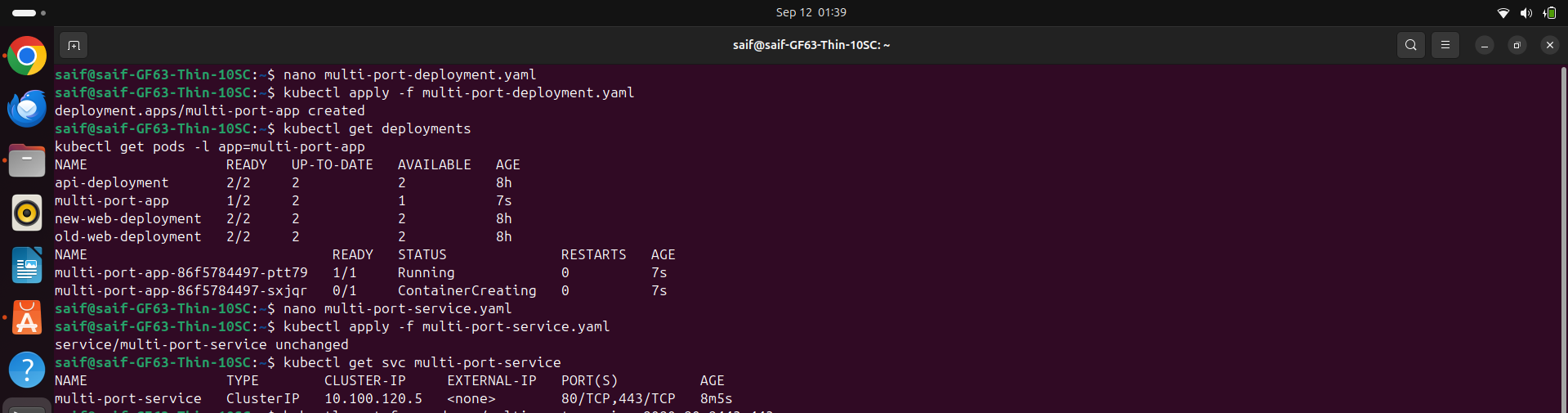


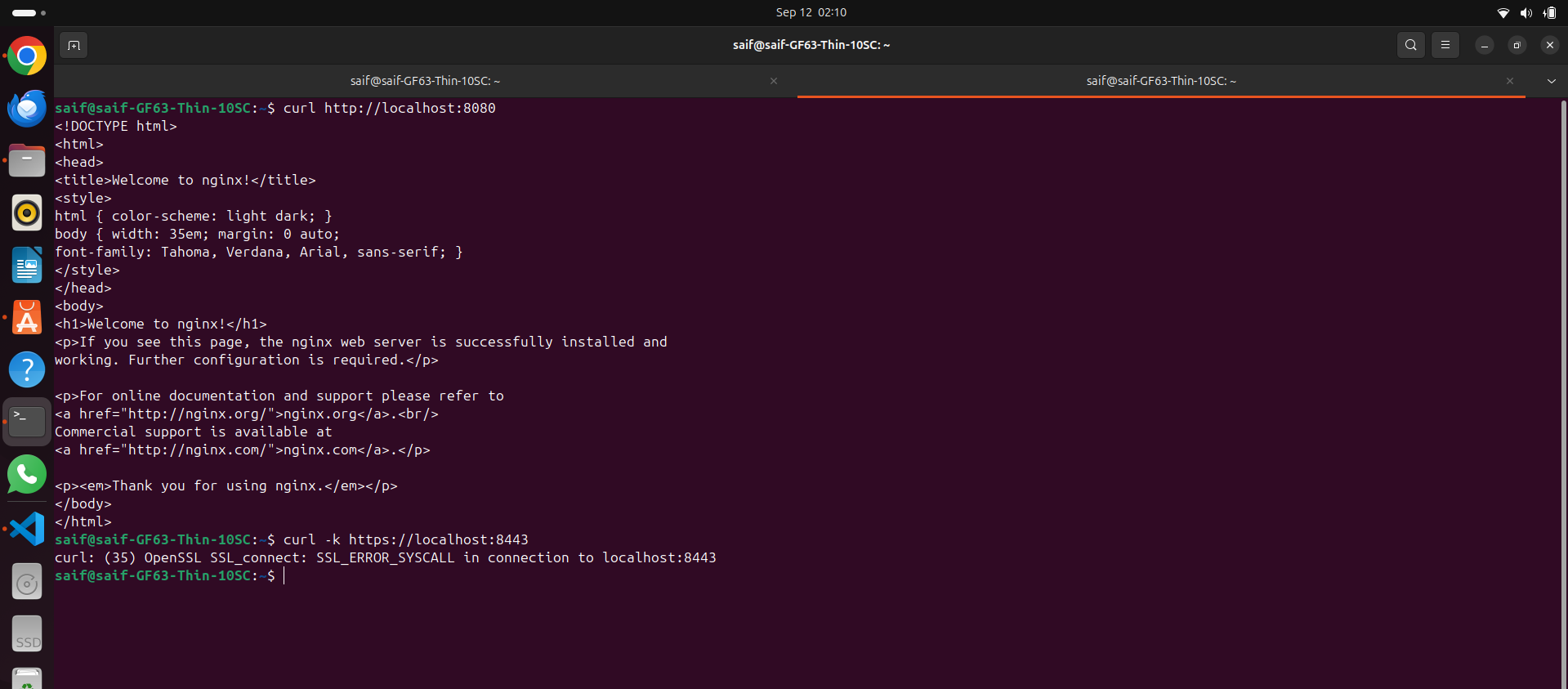
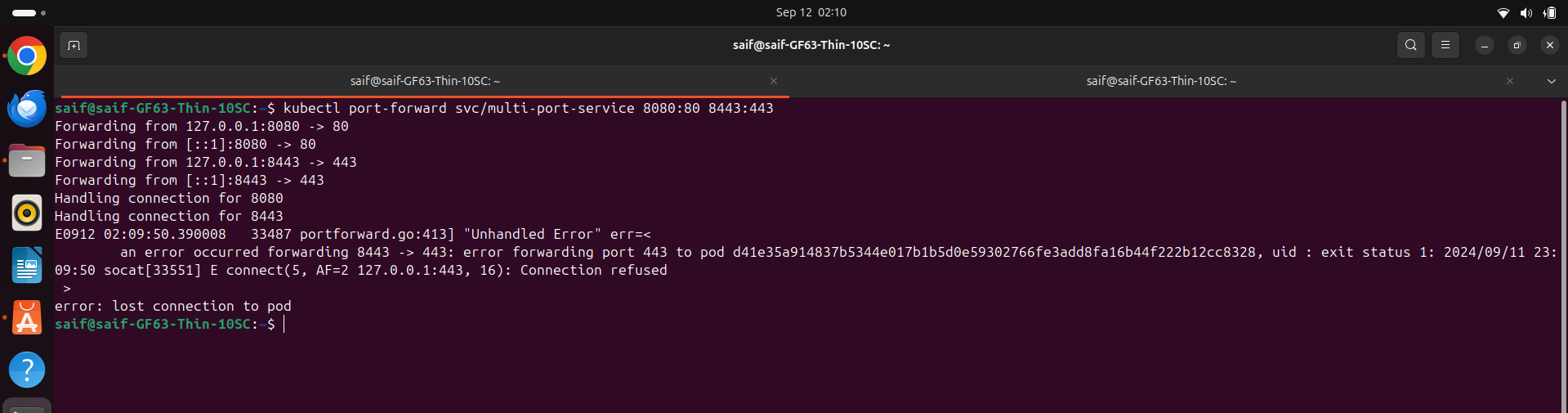


Changing from old-web to new web



- Configure a Service for Multiple Ports:  
 Define a service that exposes multiple ports (e.g., 80 for HTTP and 443 for HTTPS) on a deployment named multi-port-app "get the front in app from your own choice". Verify both ports are accessible within the cluster.



the service inside the pod is not listening on the port 433 for HTTPS. I tried to configure this error around one our.