**KUBERNETES LABELS AND SELECTORS**

**LABELS:**

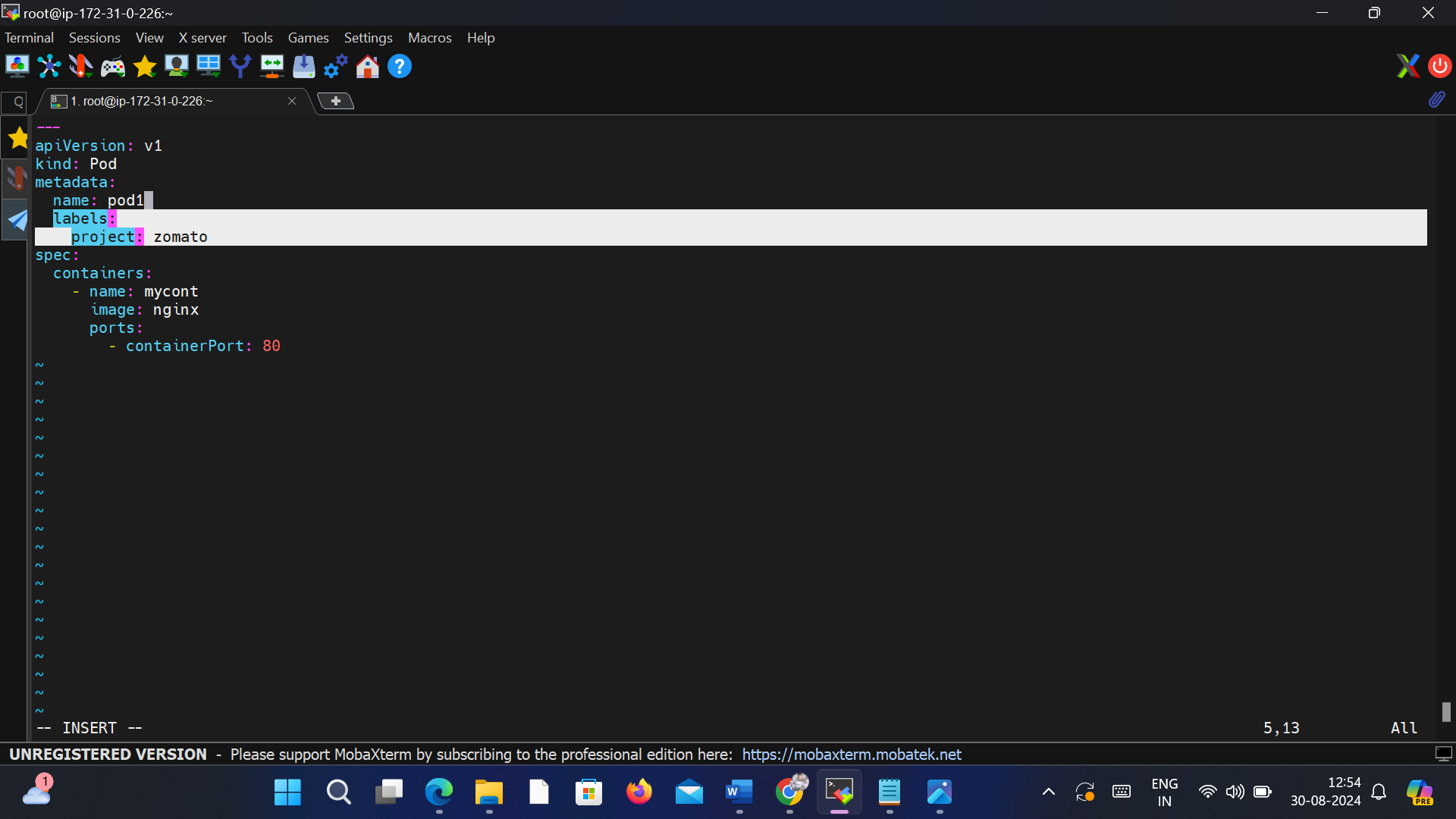
* **Labels are used to organize K8S objects such as pods,nodes etc…**
* **You can Add multiple lables over the Kubernetes objects**
* **Labels are defined in key value pair**
* **Labels are similar to tags in AWS, where you gave name to filter the resource quickly.**

**Suppose there are different pods.**

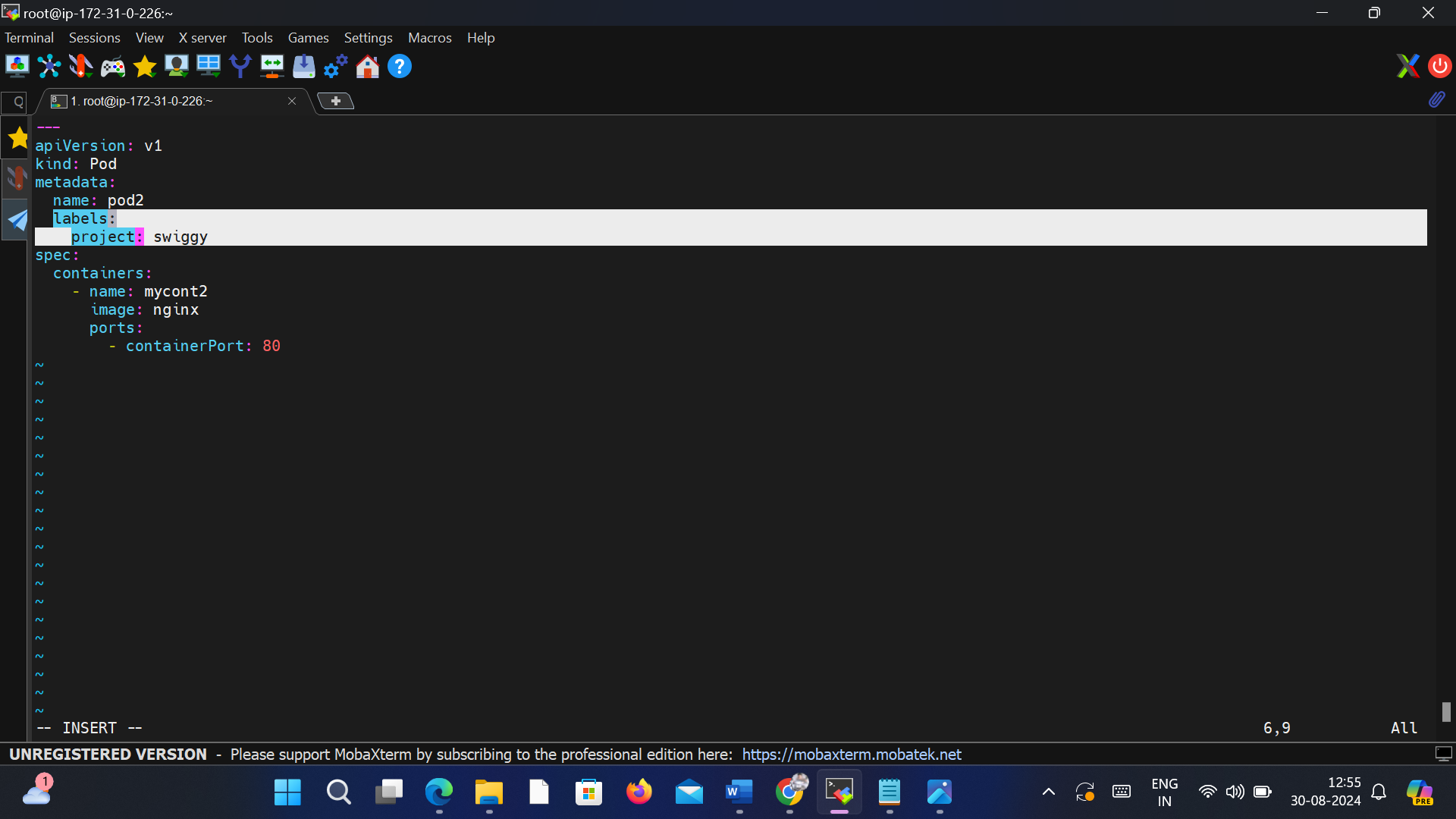
**I have created pods related to Swiggy app and Zomato app. If I want to filter particular app pod. I create labels to that pods while creating.**

**To create Label for Pod:**

**This is Pod related to Zomato**

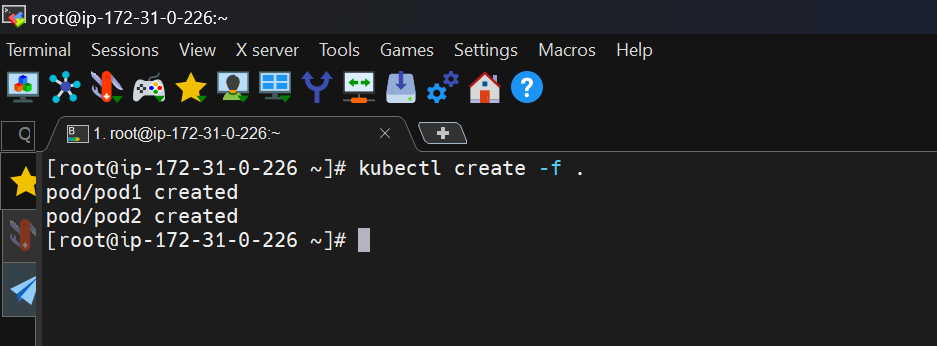


**Create a Pod related to Swiggy**



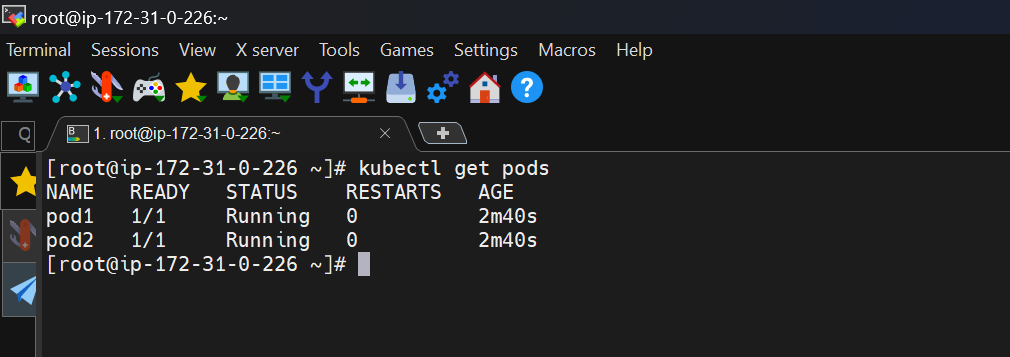
**To Create Pod**

**kubectl create -f . (By using This command created two pods at a time)**

****

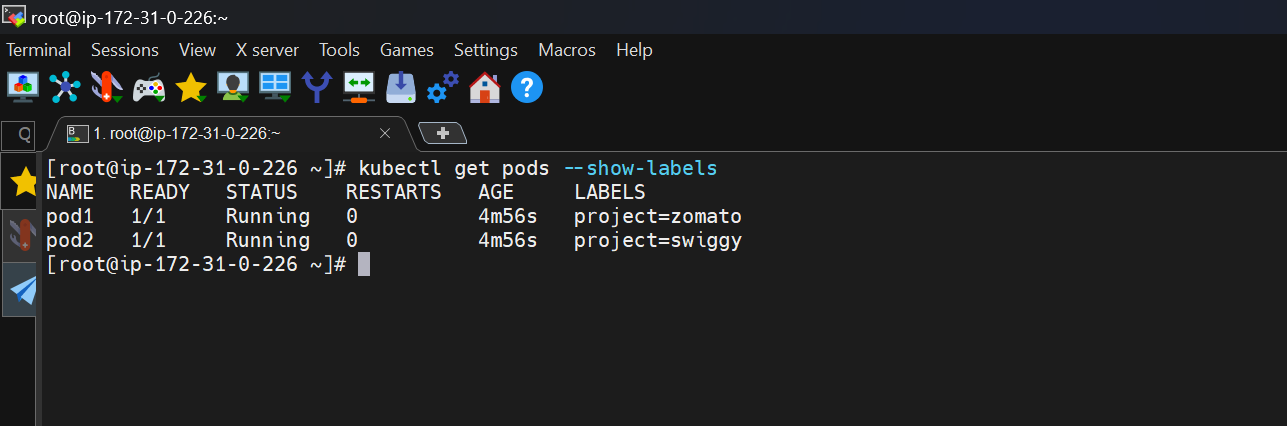
**Two Pods Created**

**kubectl get pods**

****

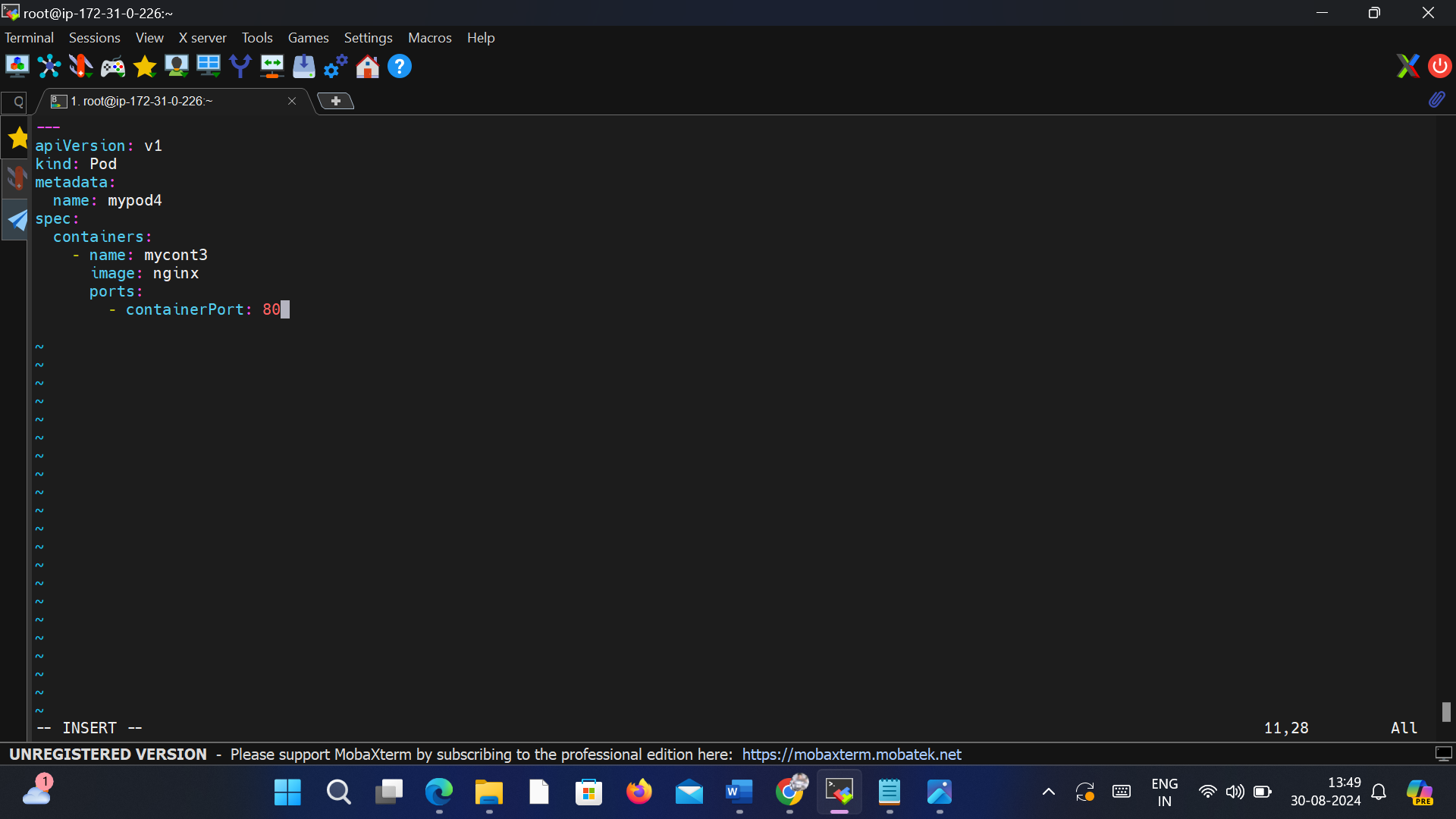
**To See labels of the pod**

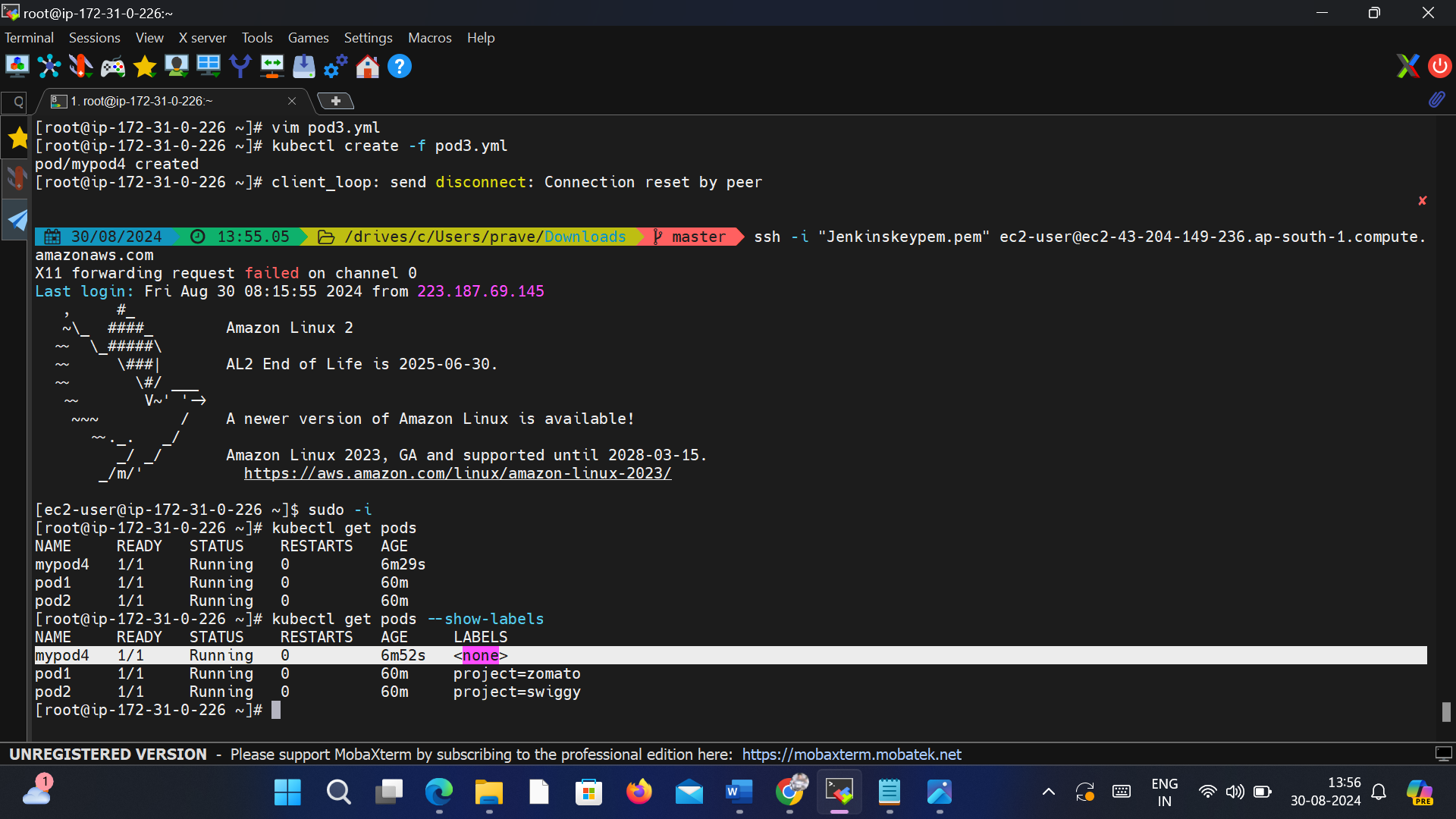
**kubectl get pods - -show- -labels**

****

**To create a labels for exsiting pod**

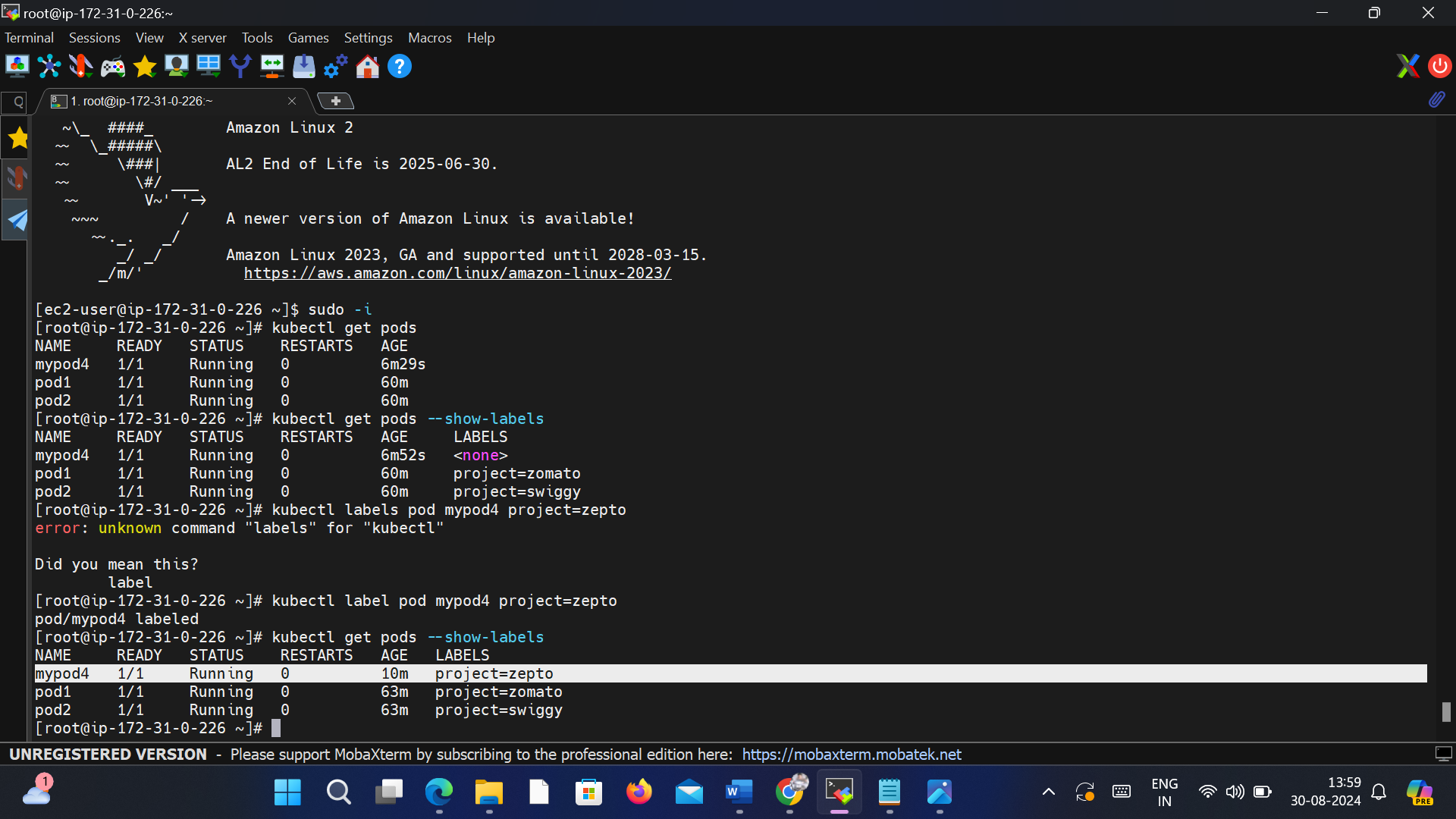
**Suppose a create a pod without labels**





**Copy the ClusterIP of service and paste in any node in cluster.We access the app.**

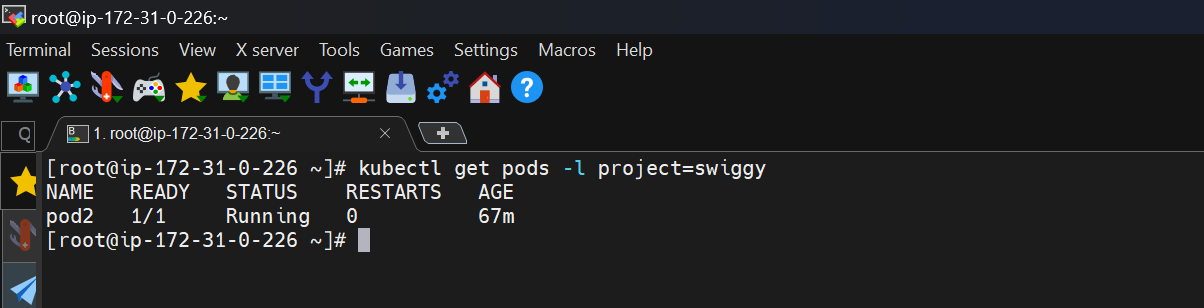
**To Create a labels for existing pod**



**To filter a pods**

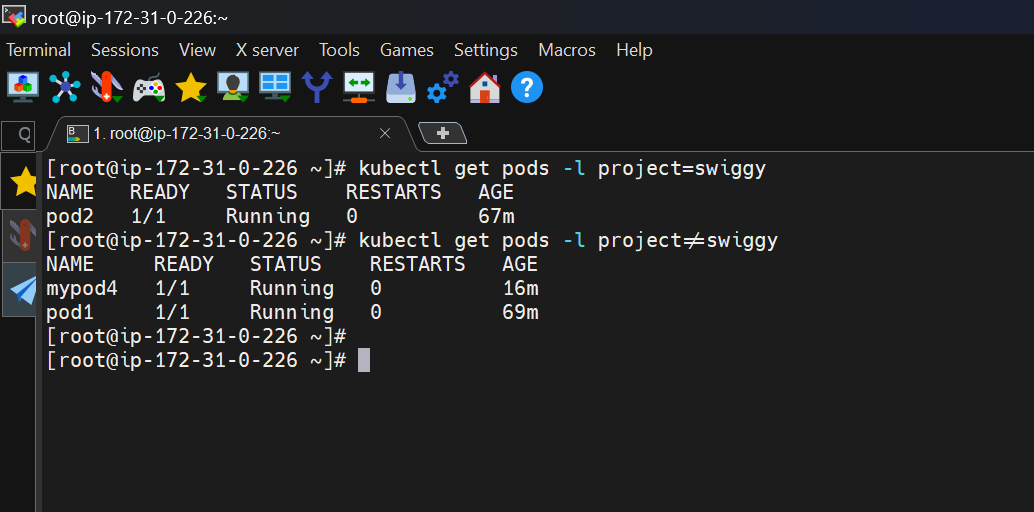
**Suppose You want to filter pods related to swiggy**

**kubectl get pods -l project=swiggy**

****

**If you want get pods except swiggy**

**kubectl get pods -l project!=swiggy**

****

**SELECTORS:**

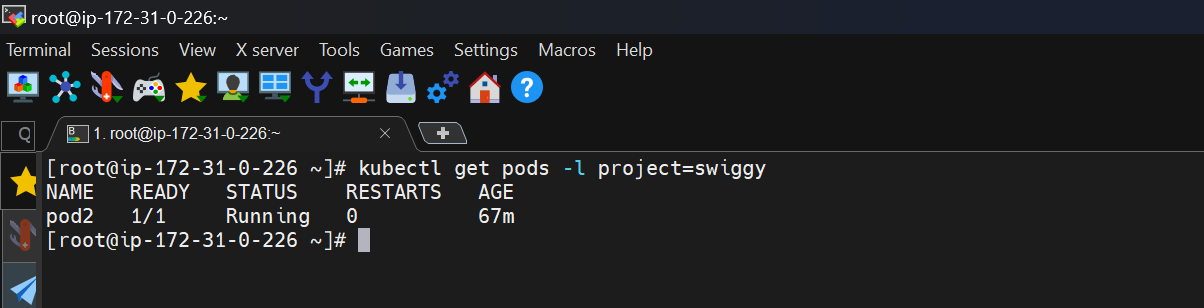
* **Selector helps us to filter the items/objects which have labels attached to them.**
* **Selectors are used to identify and select resources based on their labels.**
* **They allow you to define a set of criteria that a resource must meet to be selected. Selectors are commonly used to define the desired state of a resource or to create relationships between different resources.**

**There are different types of selectors in Kubernetes:**

**Equality-based Selectors:**

**While Selecting a label if you are using equalsto(=) or notequalsto(!=) then it is eqaulity -based selector.**

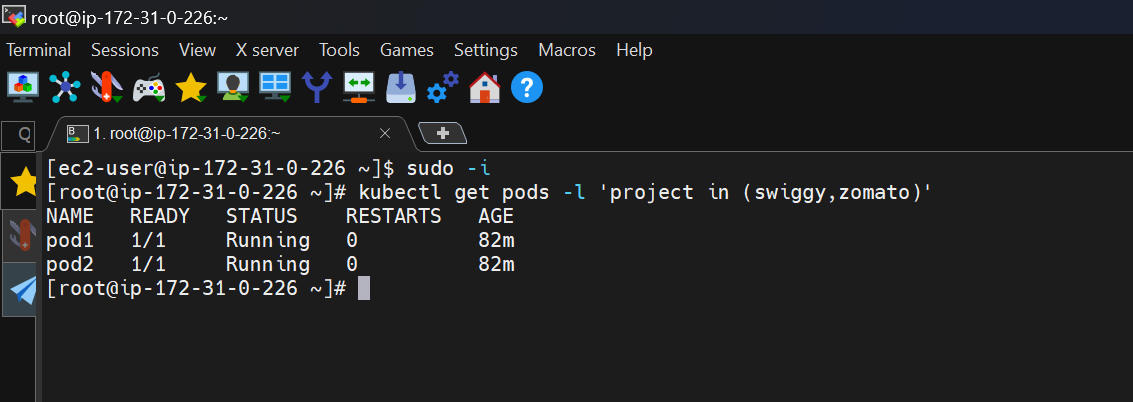
**Here we can select only single label**

****

**Set-based Selectors:**

**If you not used (=) and (!=) then it is Set-based selectors**

**Here we can select multiple labels**

****

**SERVICES**

* **Service is a method for exposing pods in your cluster**
* **Each pod gets its own IP address but we need to access from IP of the Node.**
* **If you want to access pod from inside (within the cluster) we use ClusterIP**
* **If you want to expose application to external we use NodePort and Load Balancer**

**Types**

* **CLUSTER-IP**

**ClusterIP is an internal IP type, accessible only within the cluster. This type is assigned to a service when you don’t explicitly specify the type in the YAML file. It’s suitable when your application should only be accessed within the cluster, ensuring internal communication. It’s like an internal phone line for services to communicate with each other.**

**By using this we can access app within the cluster**

* **NODEPORT**

**By using NodePort, you can access a service with the NodePort type from outside the cluster. This is achieved by sending a request to any of the nodes in the cluster using the specified node port. The node will then forward the request to the appropriate service, which in turn directs the request to the pods associated with that service. Consequently, the service becomes accessible from outside the cluster.**

**To access the app on internet we use NodePort and LoadBalancer**

* **LOAD BALANCER**

**It is an extension type of the nodeport service. When using a LoadBalancer service type, incoming traffic from the external world is directed to the load balancer's external IP. The load balancer then routes the traffic to the appropriate nodes and their respective ports based on the defined rules, typically using the NodePort allocated for the service on each node. This abstraction adds an extra layer and ensures that traffic is managed and distributed efficiently across the nodes in the cluster. Direct access to a node from the external world is not typical or intended with a LoadBalancer service type.**

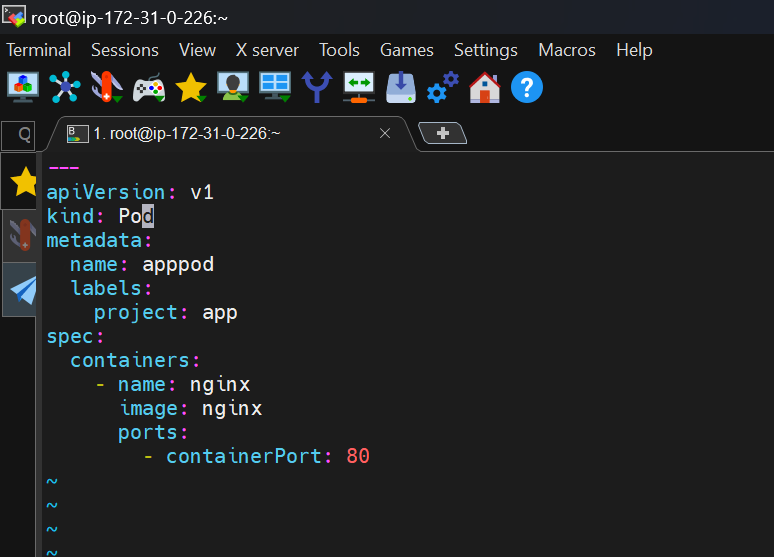
**By using LoadBalancer we can expose app to external,using cloud providers load balancer.This type of service is used when app typically handle high traffic loads**

**EXAMPLES:**

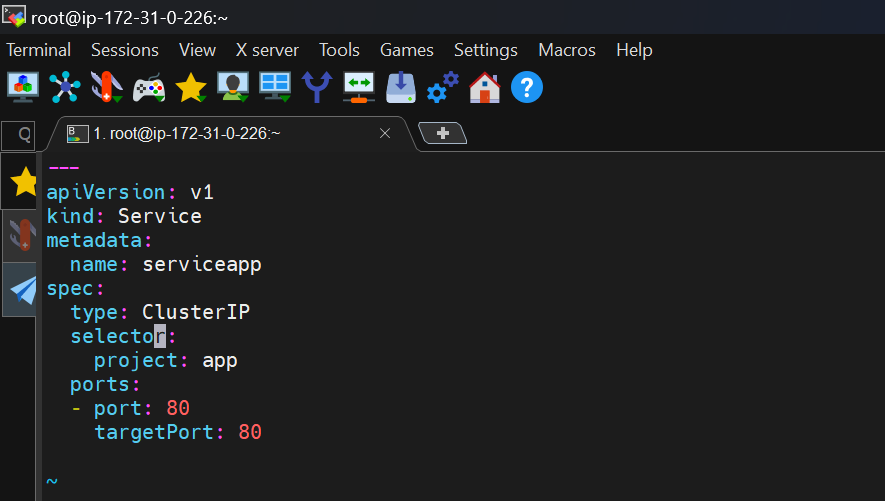
**ClusterIP**

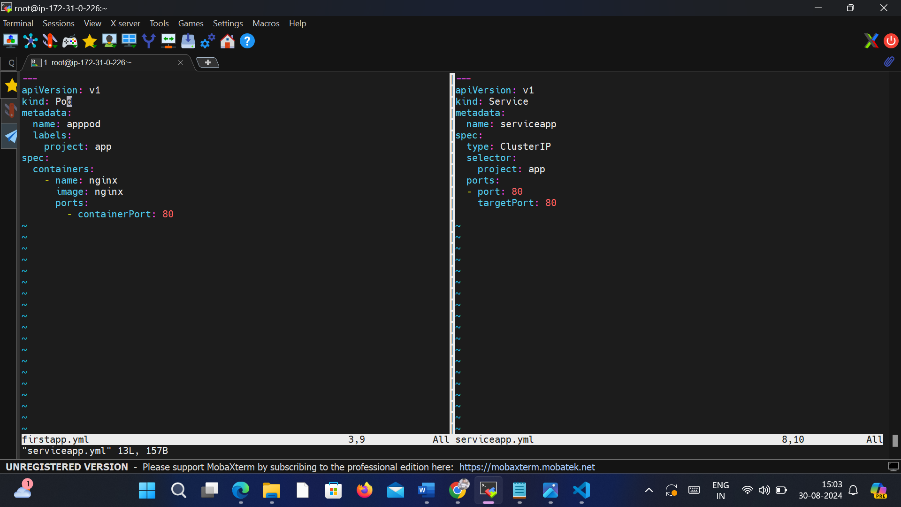
**Create two yml files**

1. **Pod yml file named as firstapp.yml**

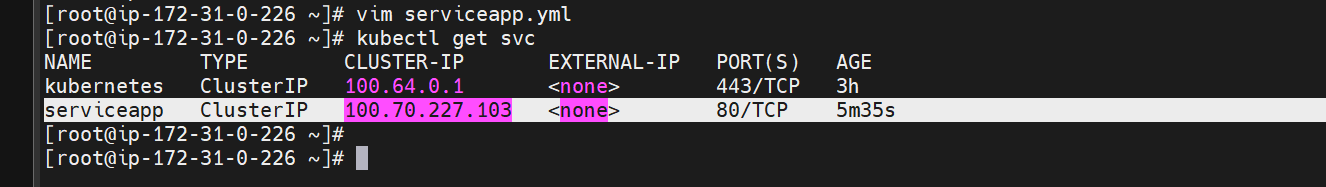


1. **Service Yaml file named as serviceapp.yml**

****

****

**In service yml file I mentioned type as ClusterIP.**

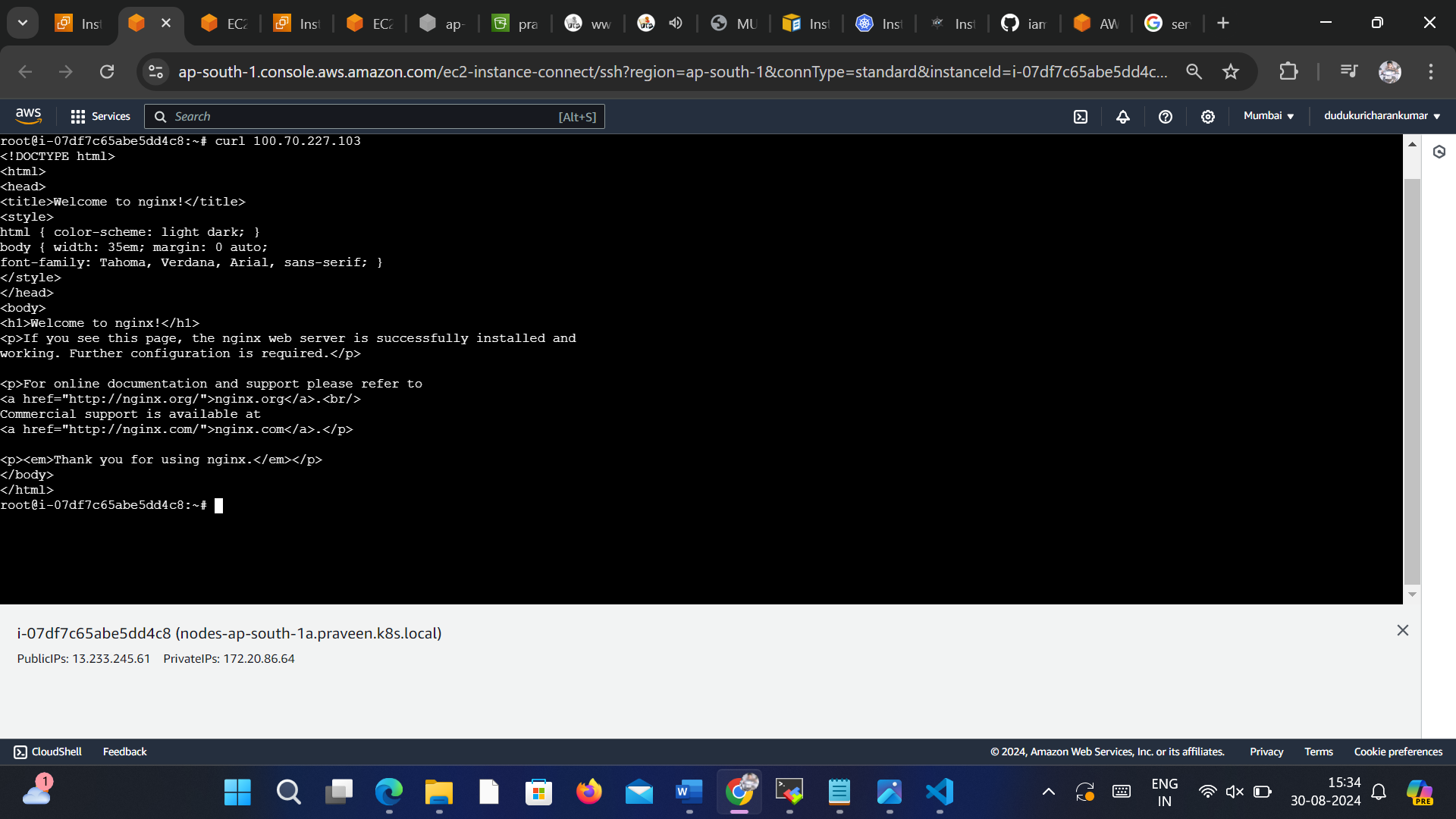
****

**By using ClusterIP we can acces app within the cluster.We can’t acces from it outside.**

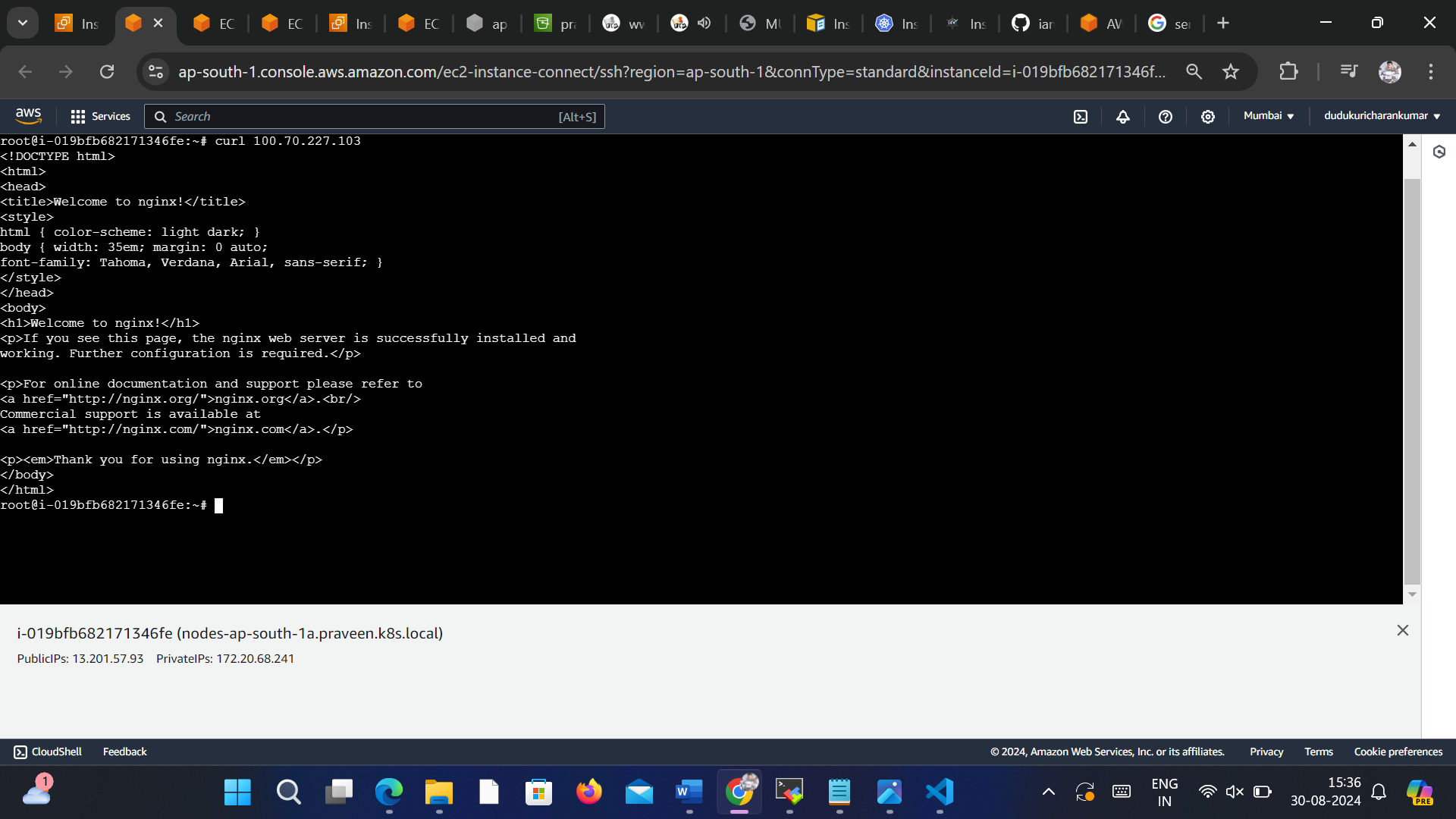
**curl cluster\_ip**

**curl 100.70.227.103**

**Node 1:**

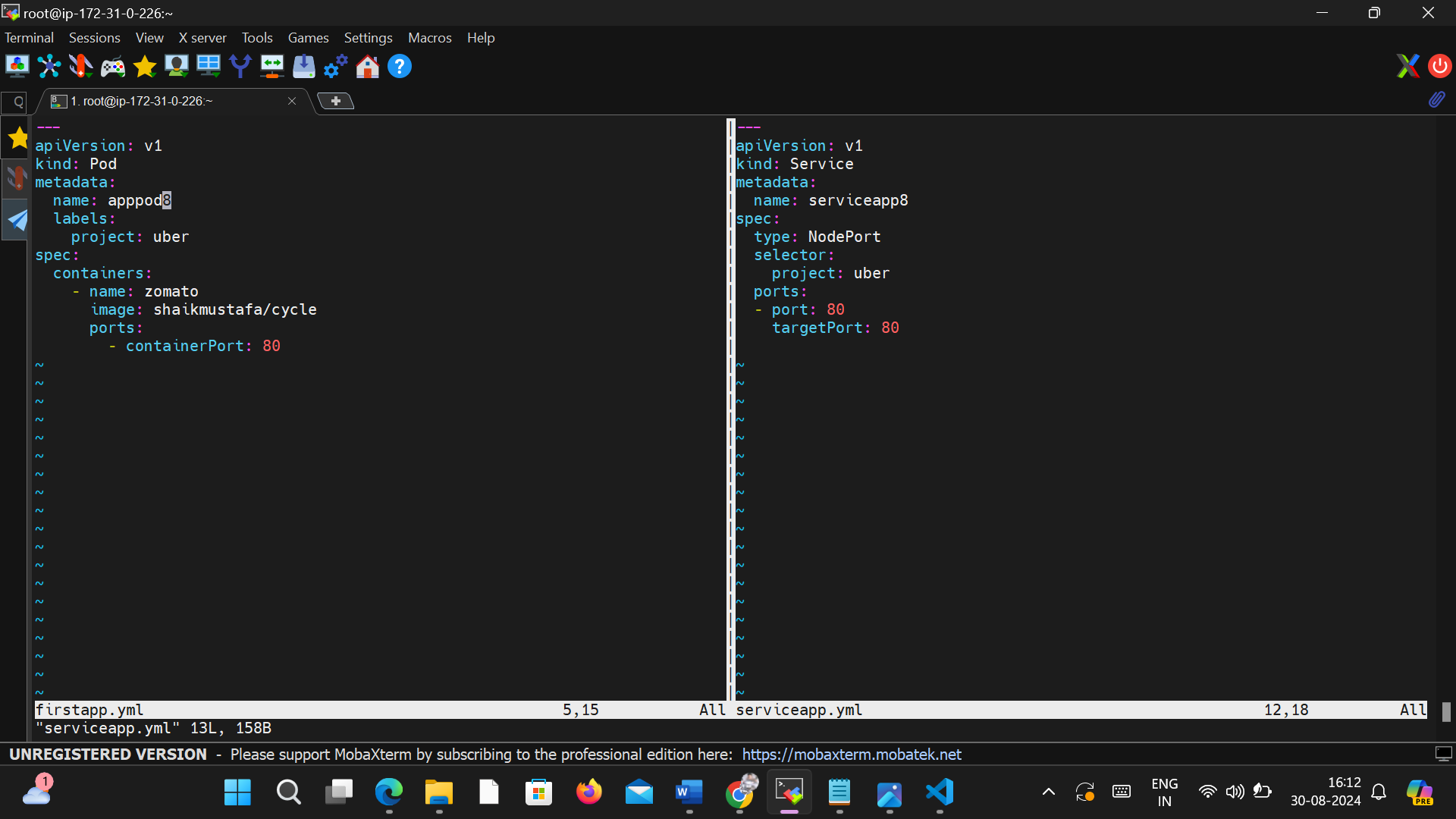


**Node 2 :**

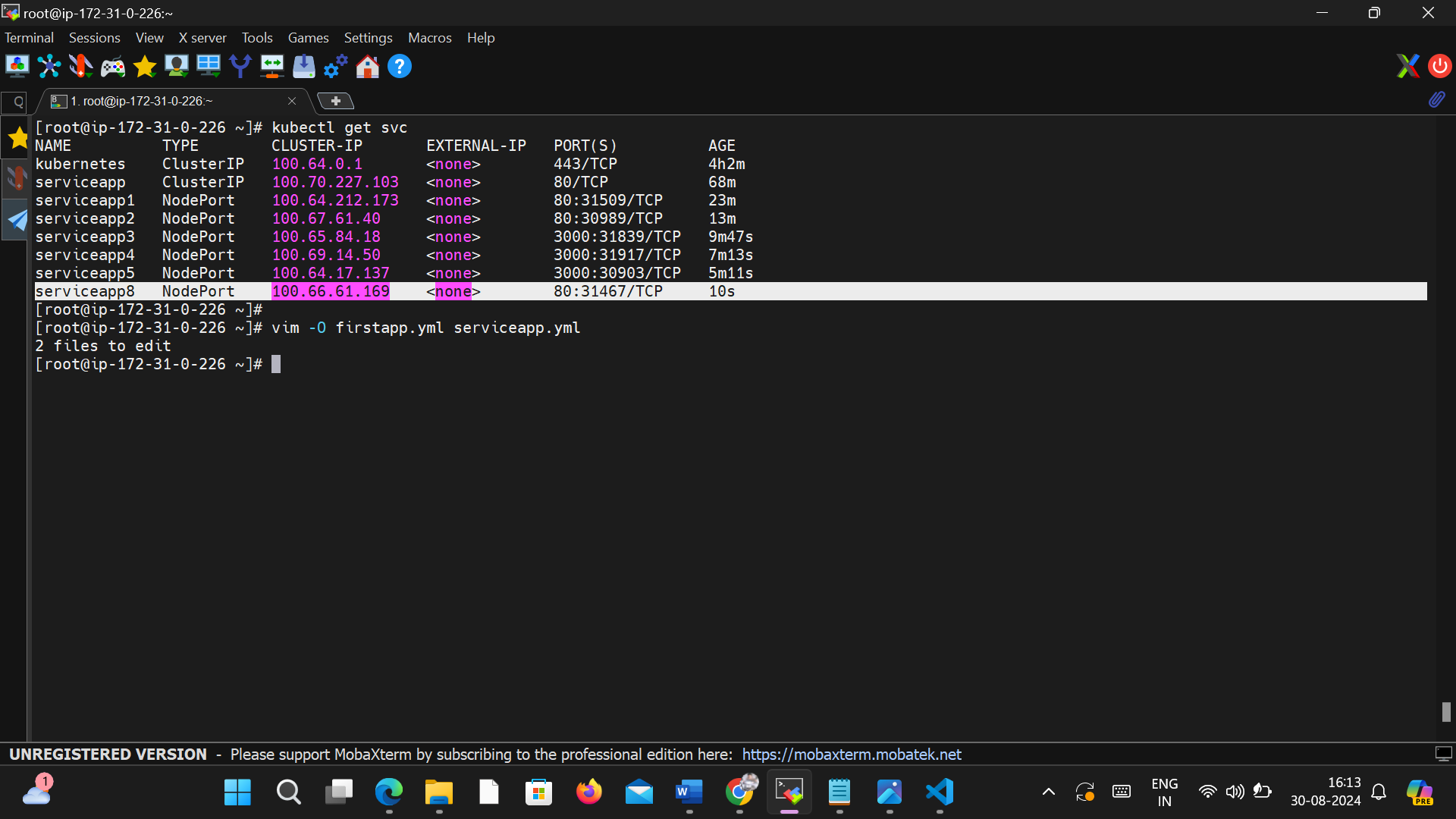


**NODE PORT:**

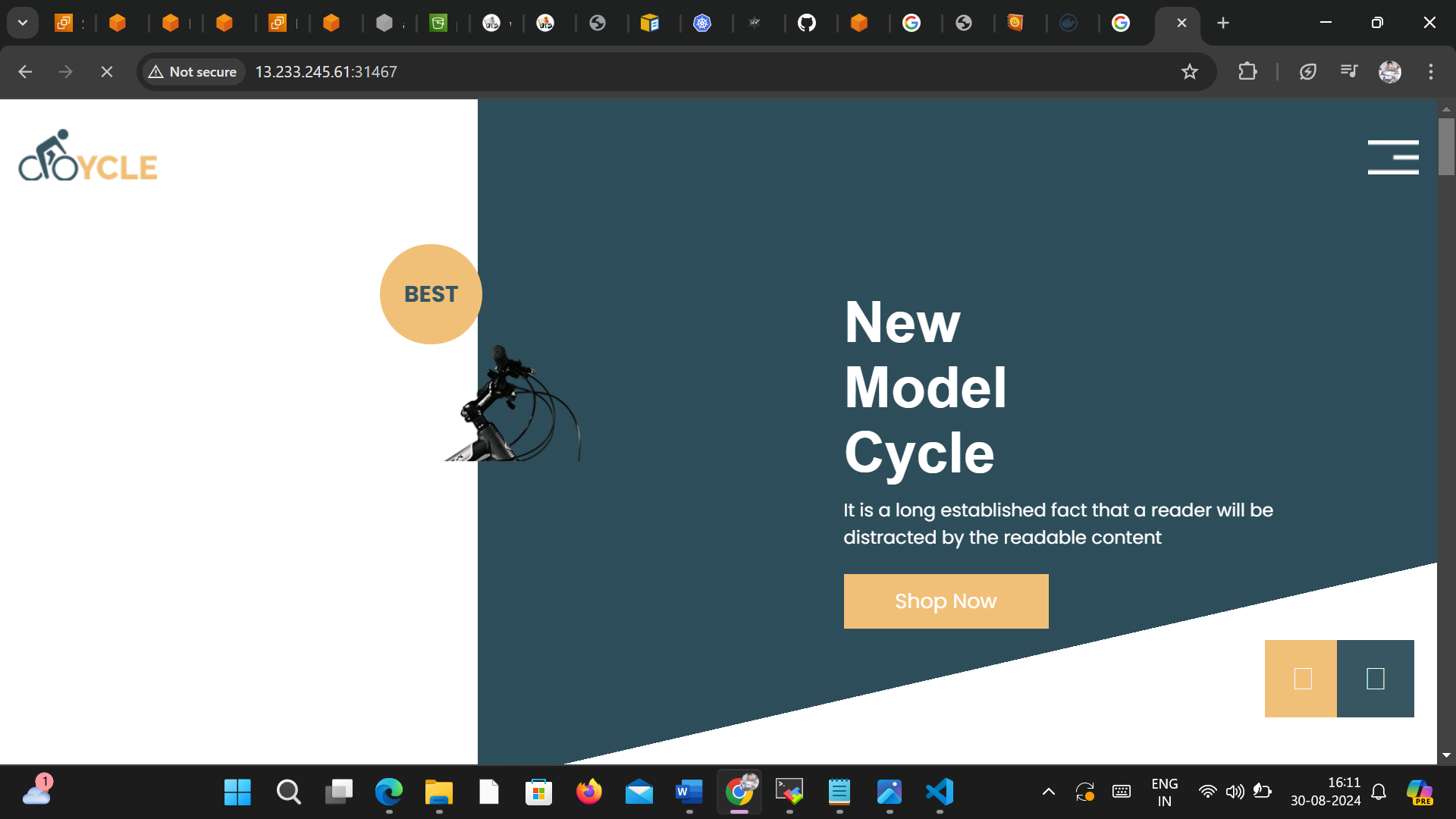
**To expose app to external we use NodePort**



**Service is created with NodePort**



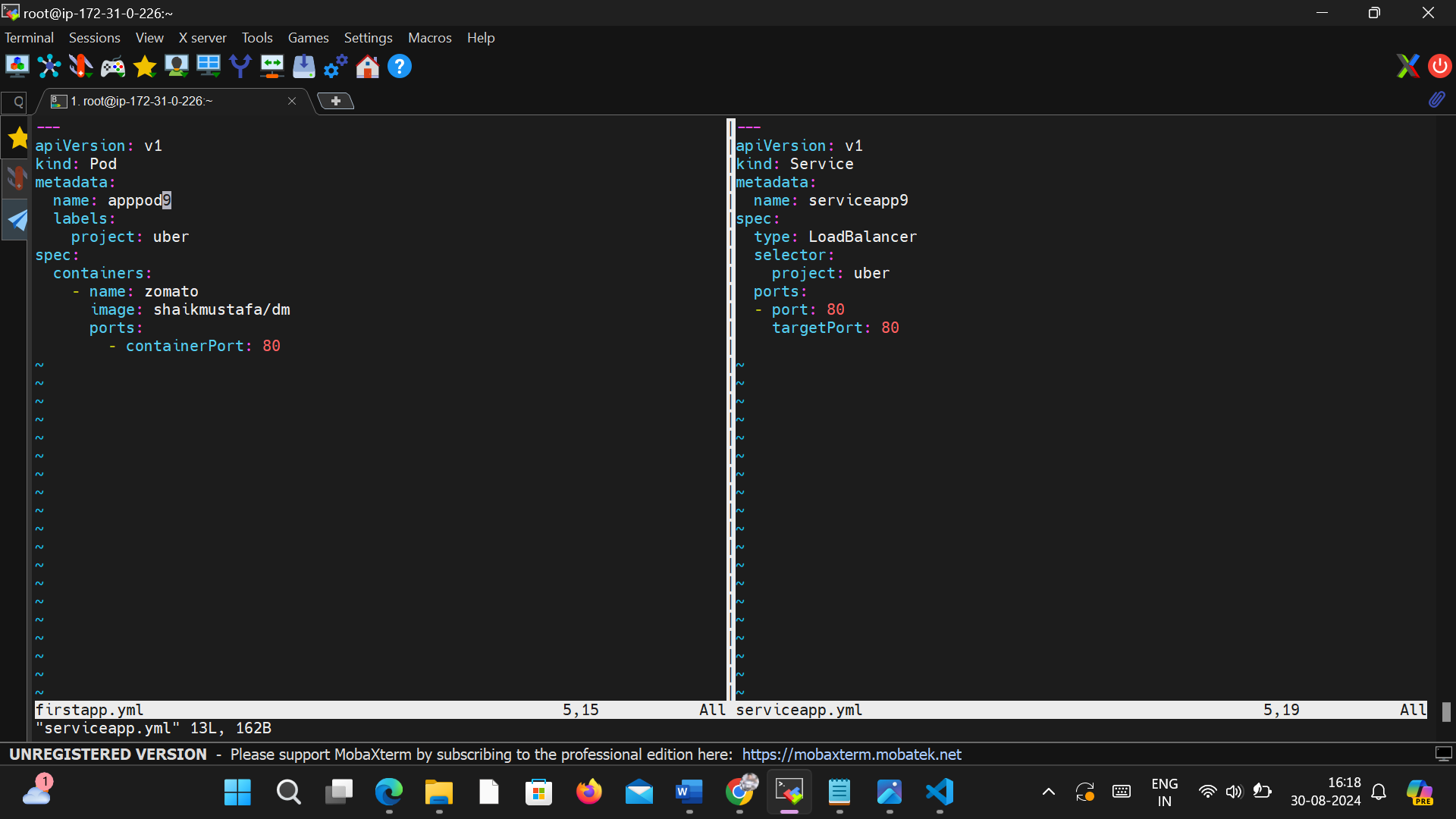
**Copy the Port of Nodeport and Paste in Browser with with public\_ip of server**

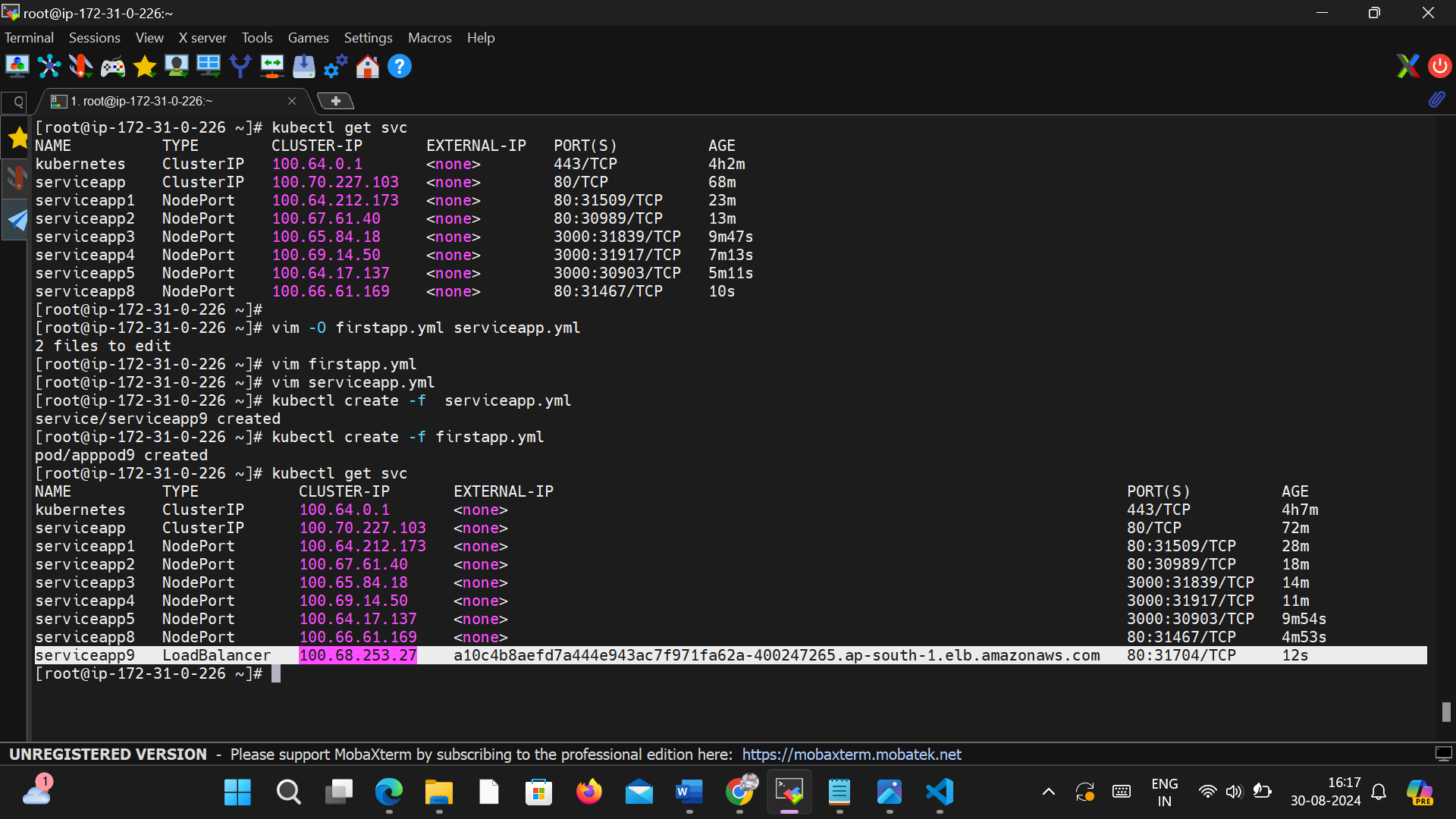


**I Successfully accessed from external (internet) by using NodePort 🥳**

**LOAD BALANCER:**

**By using LoadBalancer we can expose app to external,using cloud providers load balancer.This type of service is used when app typically handle high traffic loads**





**Copy the External IP of Load Balancer access application**

