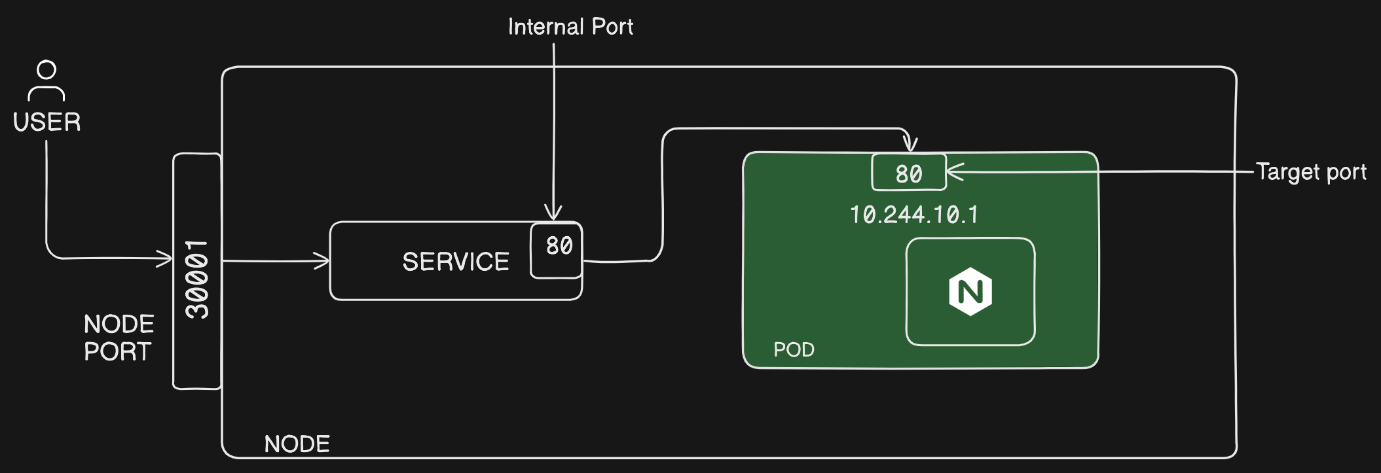
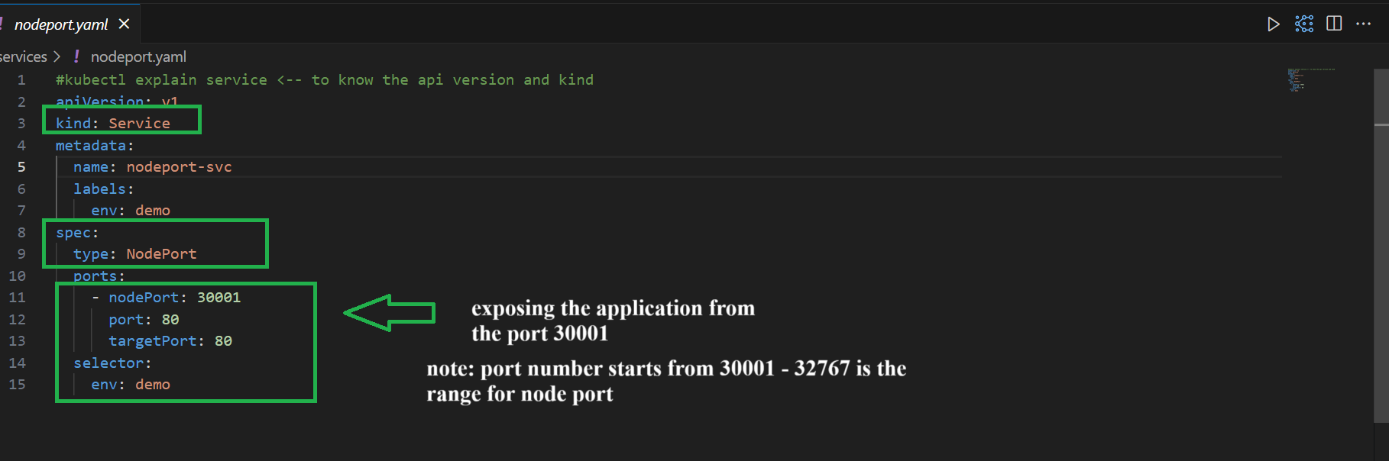
**NodePort** is a way to expose a service to external traffic. When you define a service as a **NodePort** type, Kubernetes opens a specific port on each worker node, which can be accessed from outside the cluster. This NodePort will redirect traffic to the corresponding service running inside the cluster on a specified port.

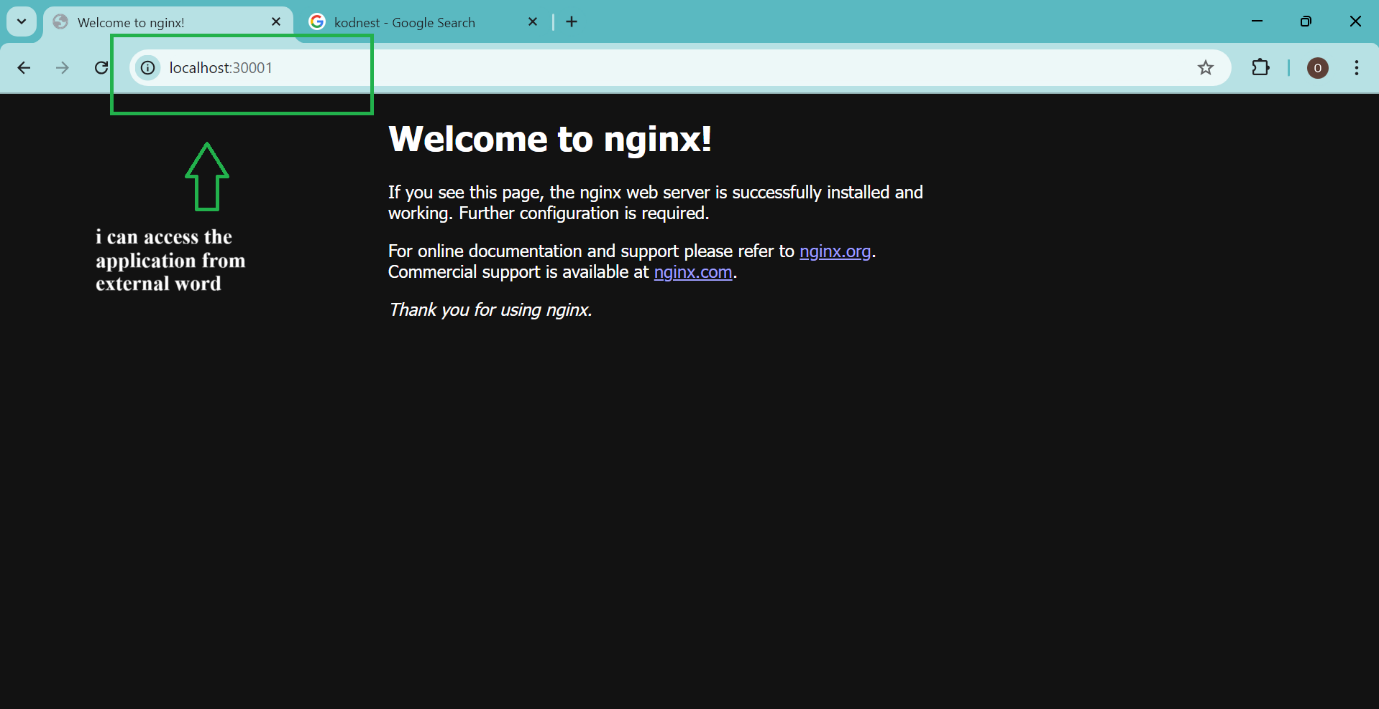


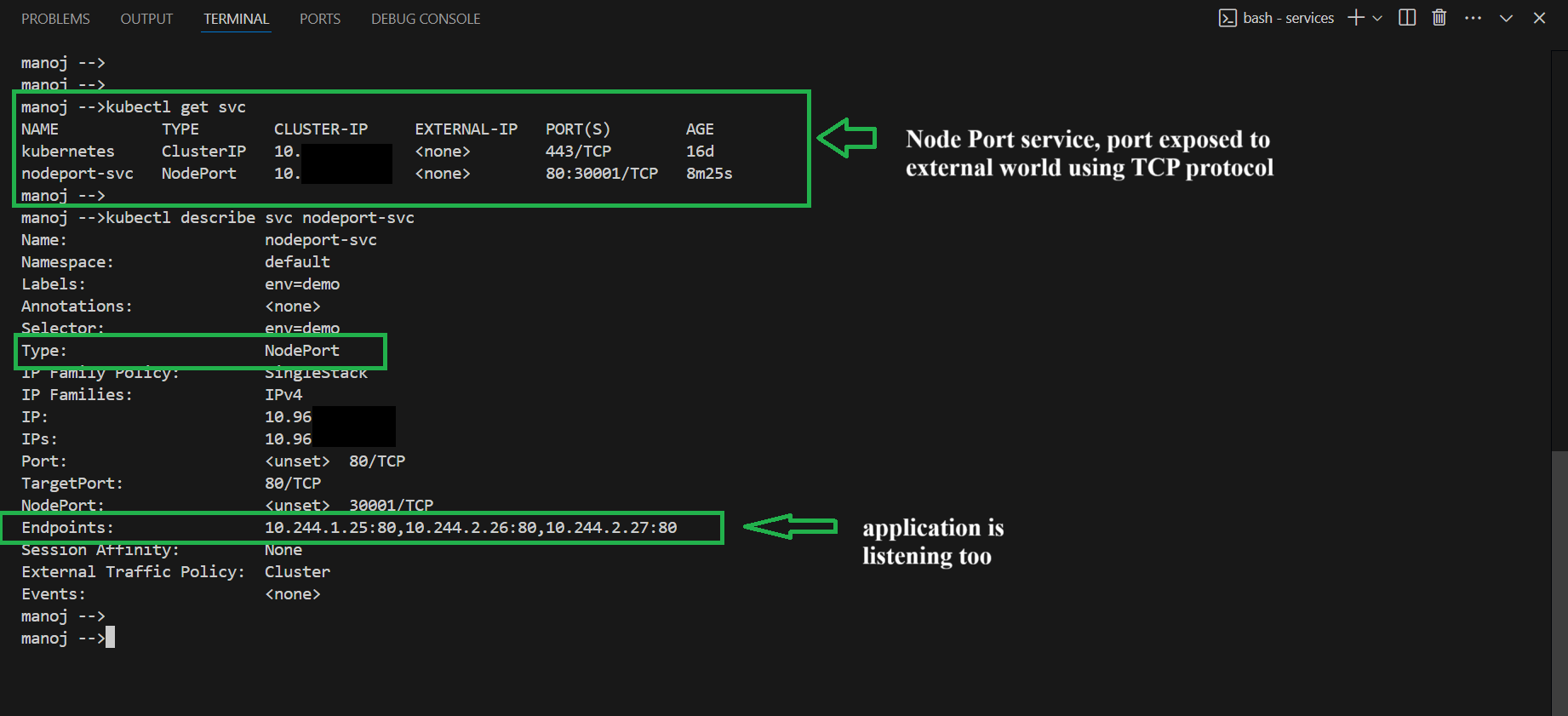
**How it works:**

1. **NodePort range**: By default, the port number is chosen from a range (usually 30000-32767), though you can specify a custom port within this range.
2. **External access**: The service becomes accessible via <NodeIP>:<NodePort>. For example, if the Node's IP is 192.168.1.10 and the assigned NodePort is 30001, you can access the service at <http://192.168.1.10:30001> or localhost:30001.
3. **Redirection**: Once the request hits the NodePort, it is forwarded to the target ClusterIP service inside the Kubernetes cluster, which in turn routes it to the appropriate pods.

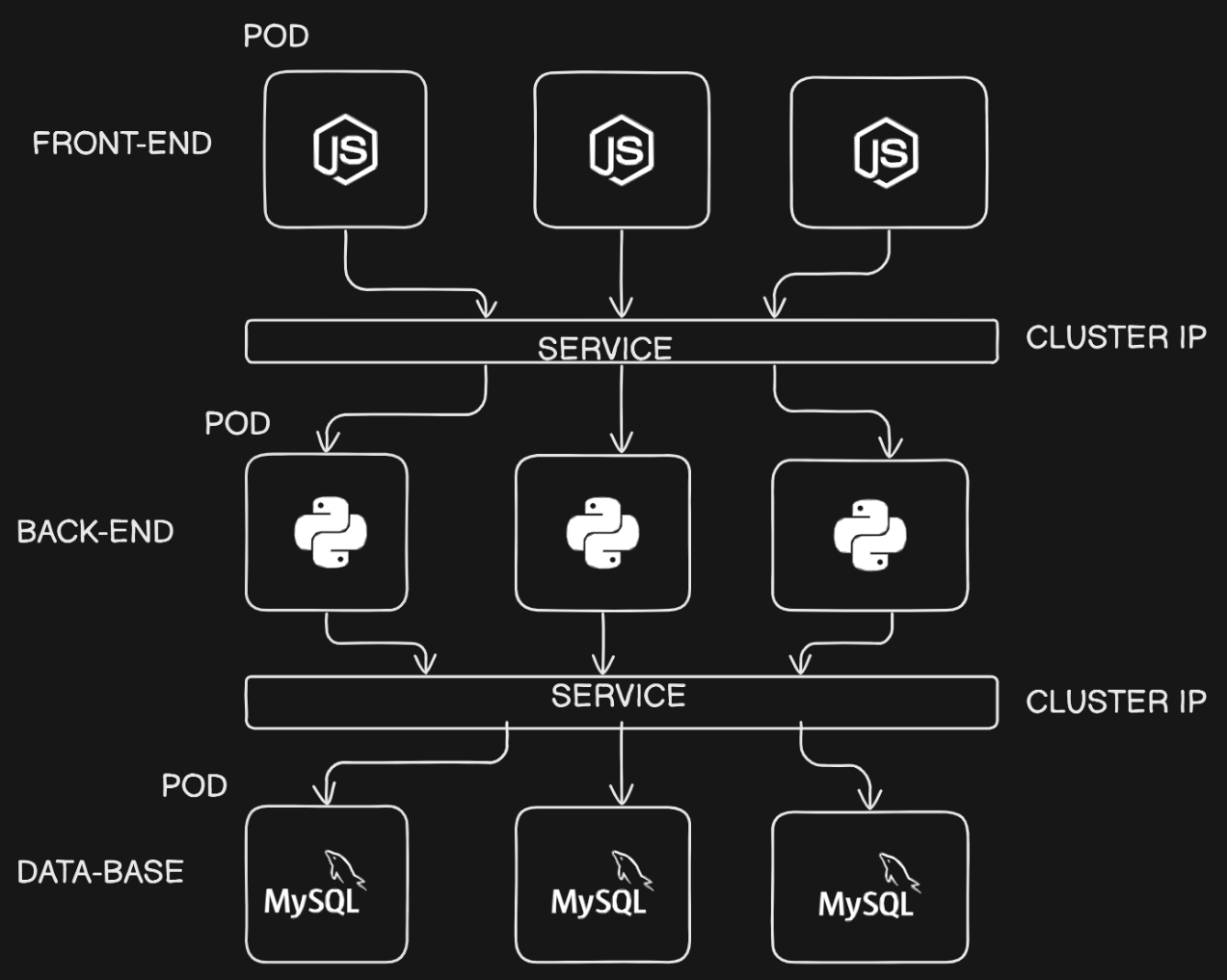
****

****

****

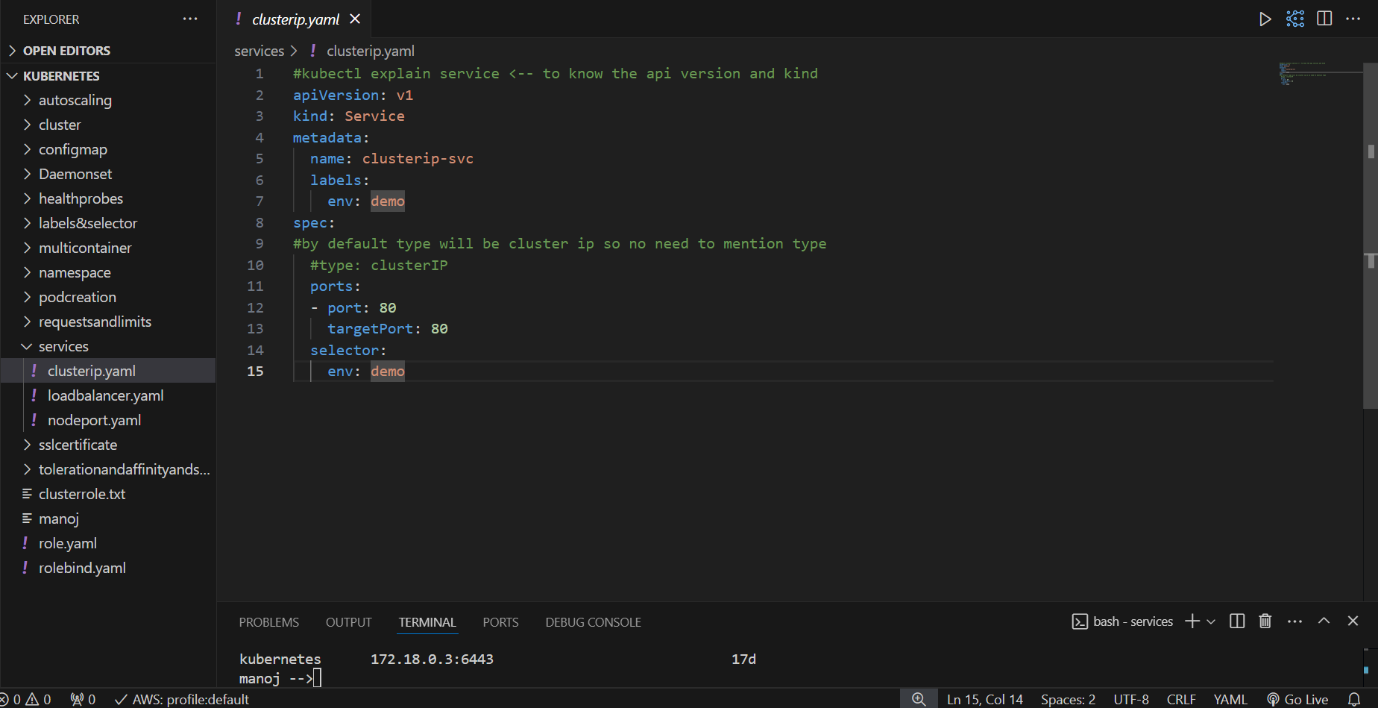
****

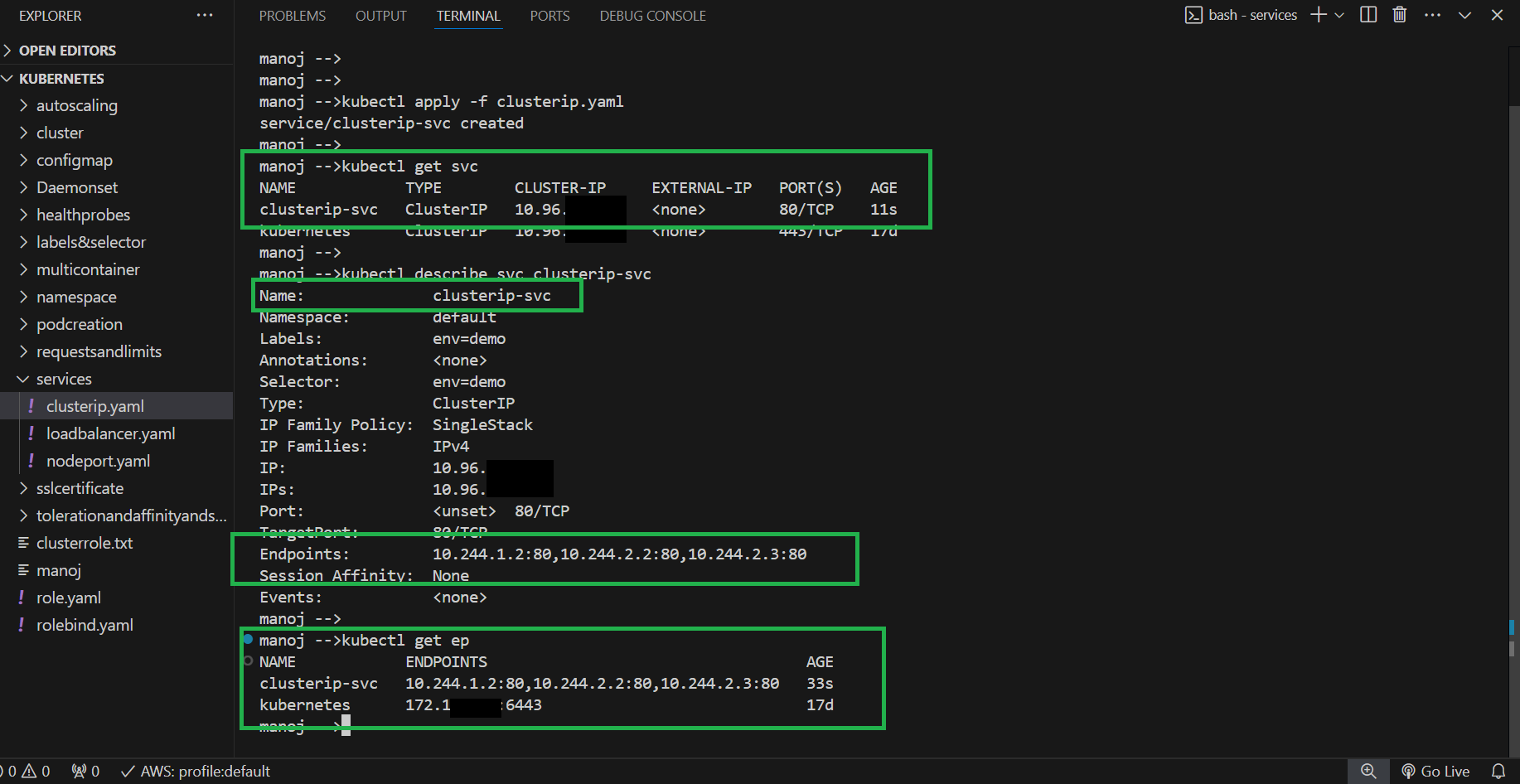
**ClusterIP** is the default service type used to expose services internally within the cluster. It provides a stable internal IP address (the ClusterIP) for communication between different components (like Pods) inside the Kubernetes cluster, without exposing the service to external traffic.

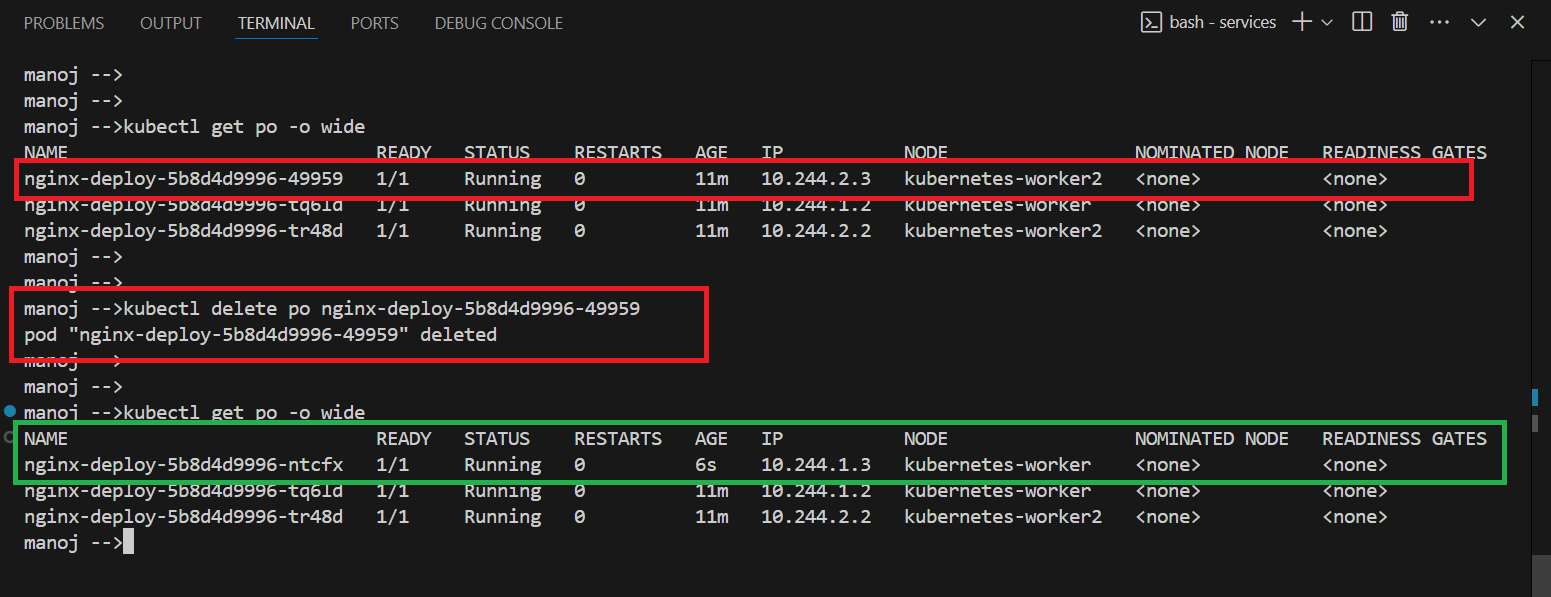


**How ClusterIP Works:**

1. **Internal Load Balancer**: The ClusterIP acts as an internal load balancer that distributes incoming requests across the Pods associated with the service.
2. **DNS Integration**: Kubernetes automatically assigns a DNS name for the service, such as my-clusterip-service.default.svc.cluster.local, which makes it easy for Pods to reference services by name instead of IP.
3. **Pod to Pod Communication**: If multiple Pods need to communicate within the cluster, they can use the ClusterIP service to reach each other without needing to know the individual Pod IP addresses.







**Key Characteristics of a ClusterIP Service:**

1. **Internal Access Only**:
   * The service is only accessible from within the cluster. It cannot be accessed from outside the cluster unless combined with other service types like **NodePort** or **LoadBalancer**.
2. **Stable IP Address**:
   * Kubernetes assigns a static IP (ClusterIP) to the service, which remains the same throughout the lifecycle of the service. This allows other Pods to communicate with the service using this internal IP address.
3. **Service Discovery**:
   * The ClusterIP is also registered with Kubernetes DNS. Pods can access the service using the service name, which simplifies service discovery within the cluster.
4. **Traffic Routing**:
   * The service acts as a load balancer that forwards requests to one of the backend Pods (selected based on labels) listening on the specified target port.