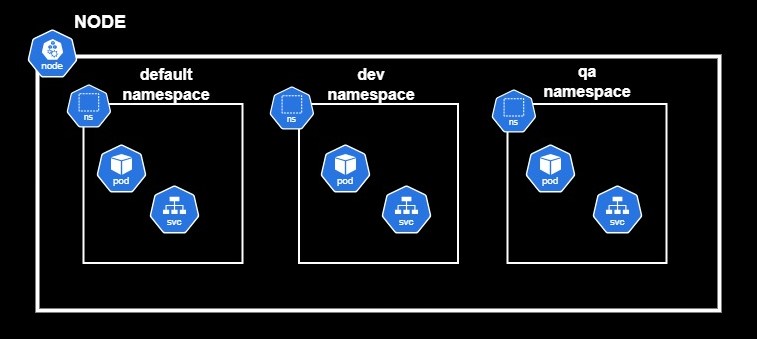
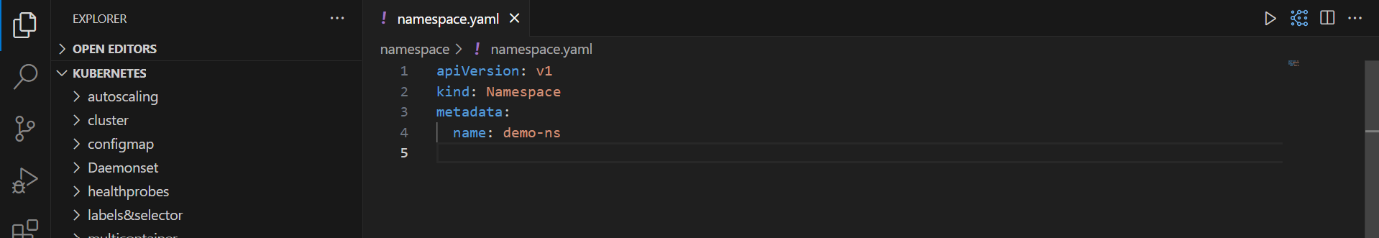
**Namespace** is a logical partition that provides a way to divide cluster resources between multiple users or teams. Namespaces are useful for organizing resources within a cluster and can be used for various purposes, such as:

1. **Isolation**: Namespaces help isolate resources (like pods, services, and deployments) within the same cluster. This is particularly useful in multi-tenant environments.
2. **Resource Quotas**: You can set resource limits and quotas on a per-namespace basis to prevent any single team or application from consuming too many resources.
3. **Environment Separation**: Namespaces can represent different environments (e.g., development, testing, production), making it easier to manage configurations and resources specific to each environment.
4. **Access Control**: Role-Based Access Control (RBAC) can be applied at the namespace level, allowing you to control who can access and modify resources within a namespace.



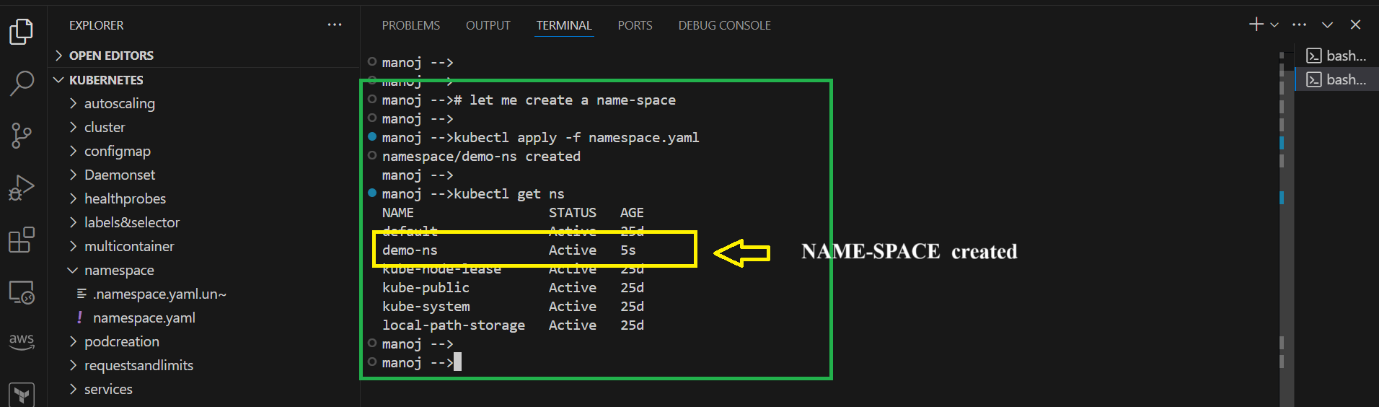
Note:

* Pod needs to communicate with in same namespace it can easily interact using “host-name”
* Pod needs to communicate with pod present in different namespace (eg. Dev, QA), we need to use “fully qualified domain name” [ FQDA].



Kubectl apply -f namespace.yaml (or) kubectl create namespace <namespace-name>

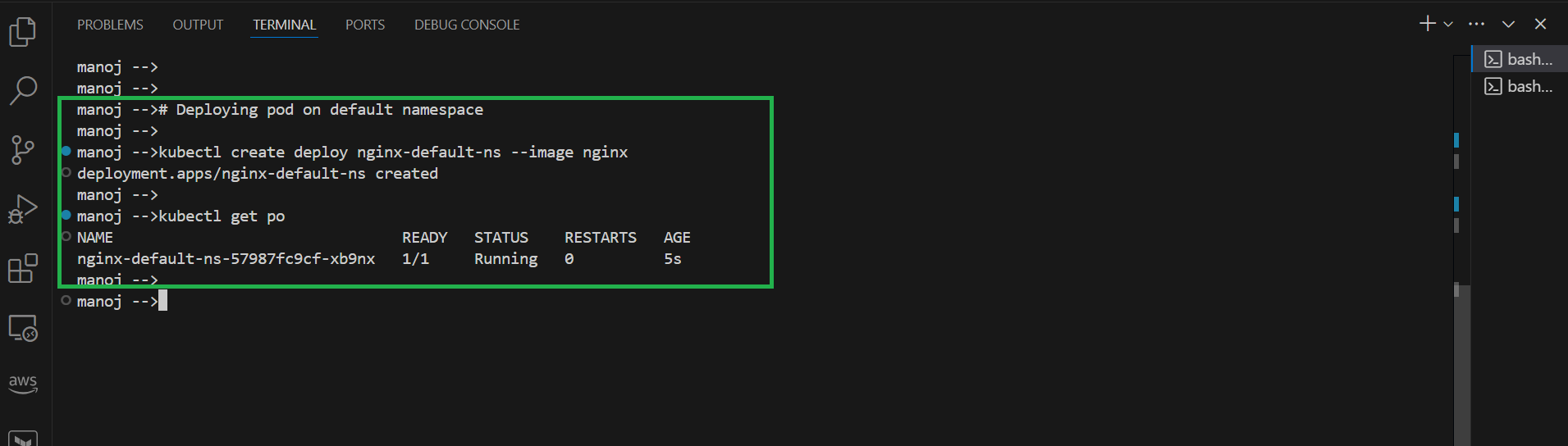
Kubectl get ns

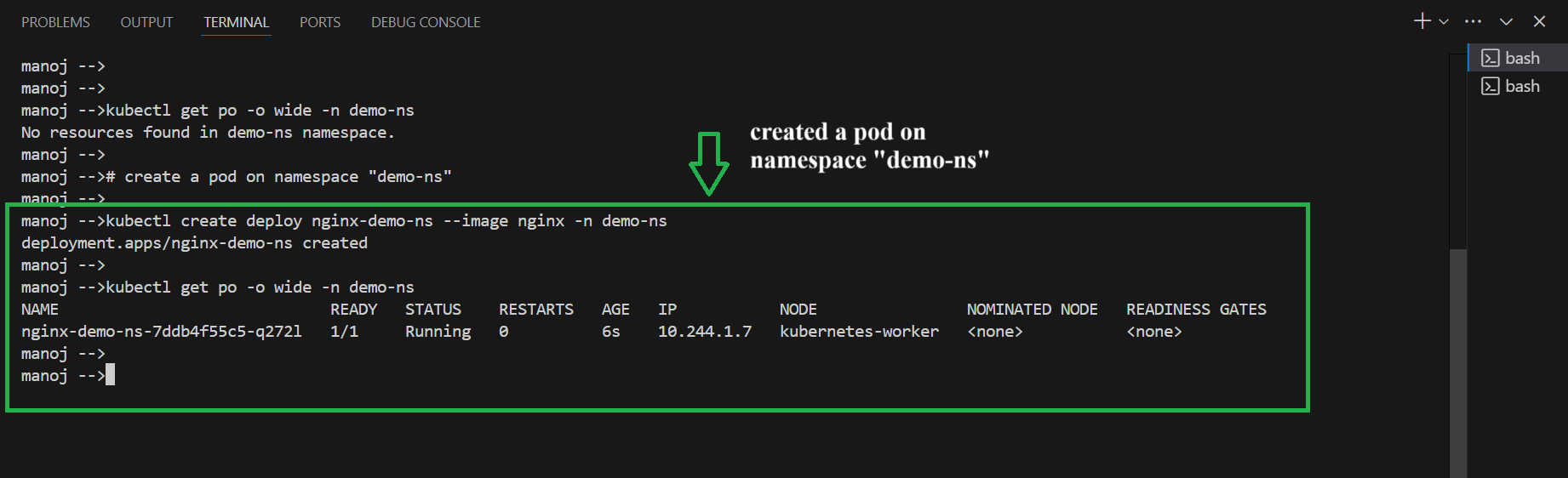


Now I want to create a pod inside namespace

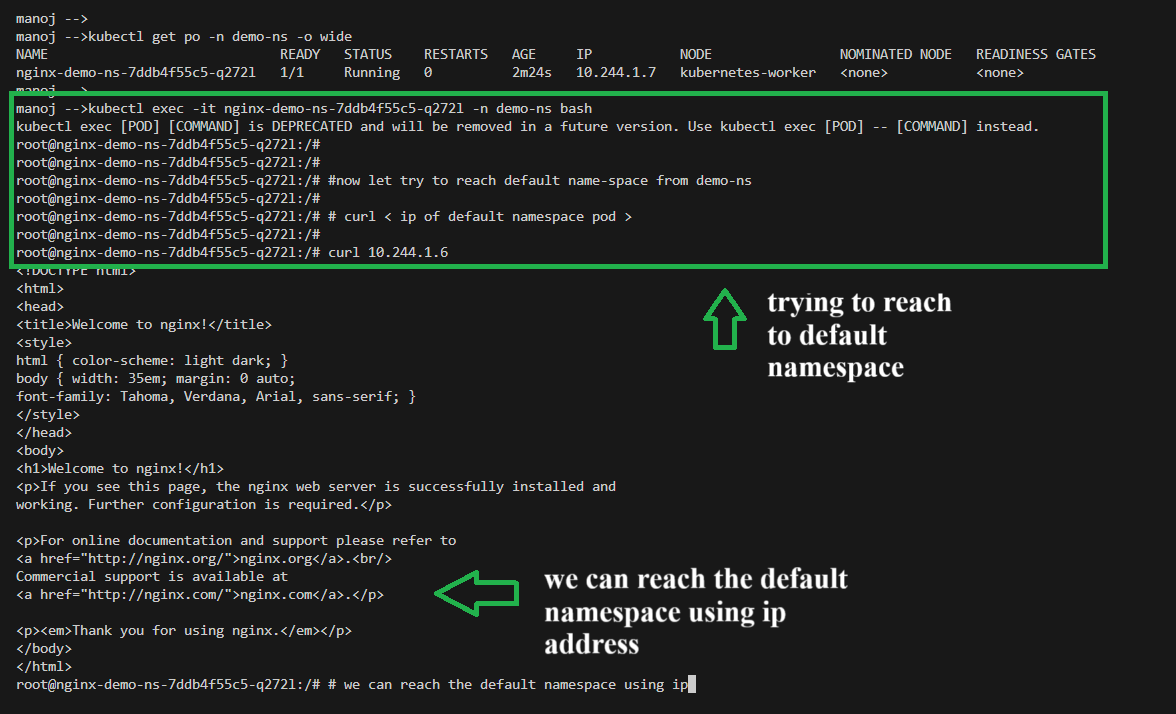
Kubectl create deploy <name of pod> --image <image-name> -n < namespace>

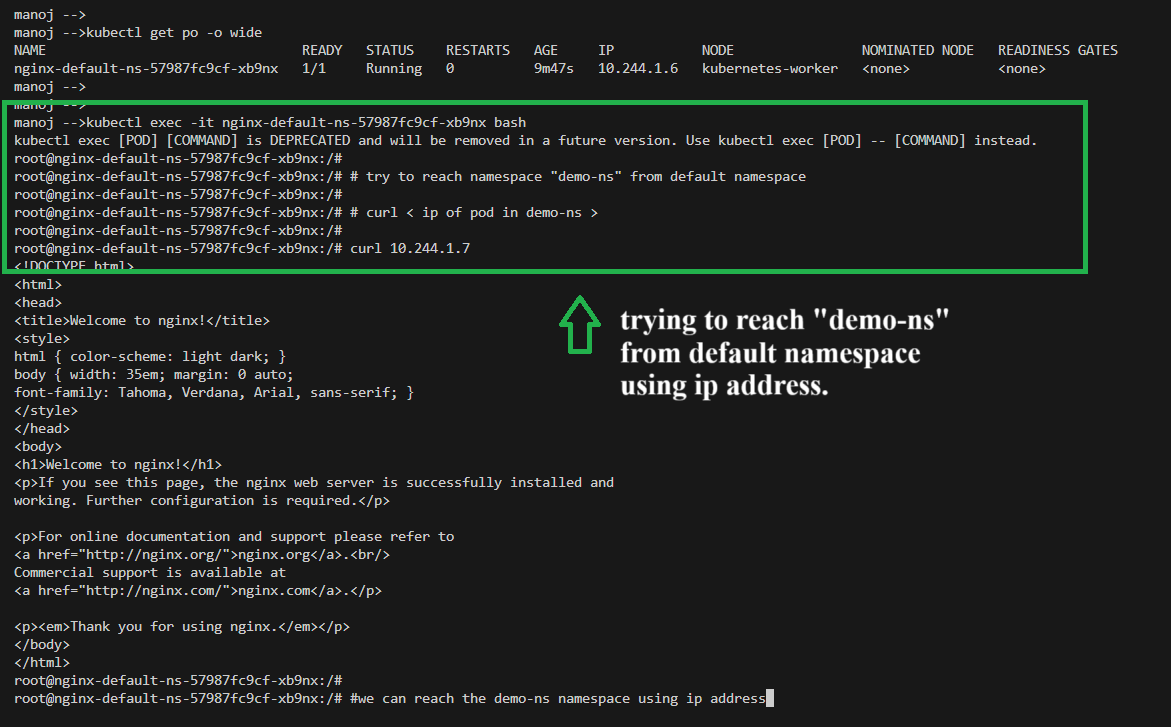
Kubectl get pod -n <namespace>



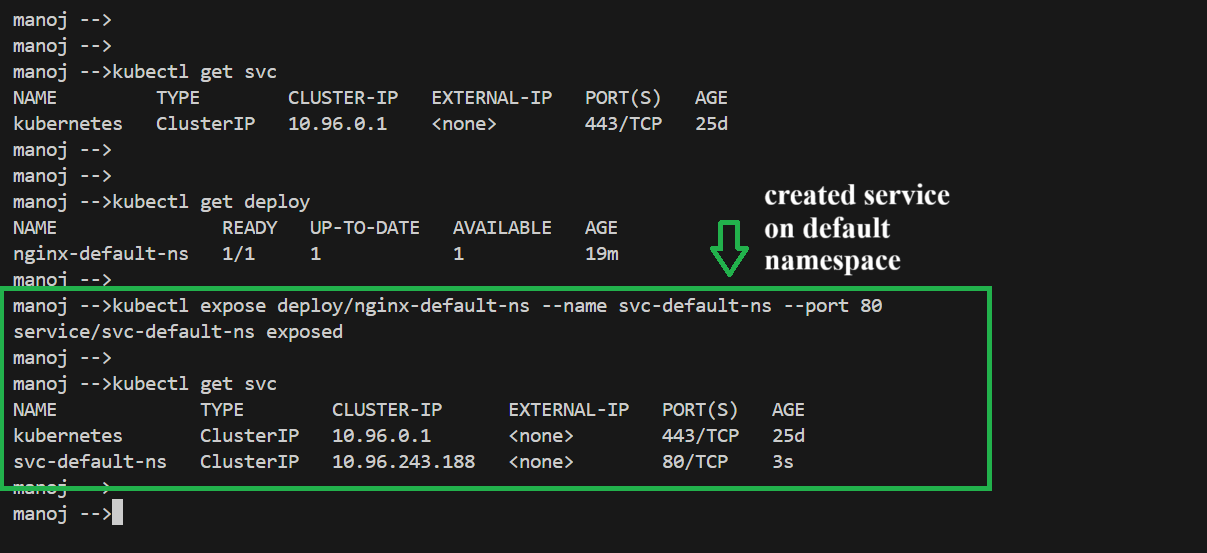


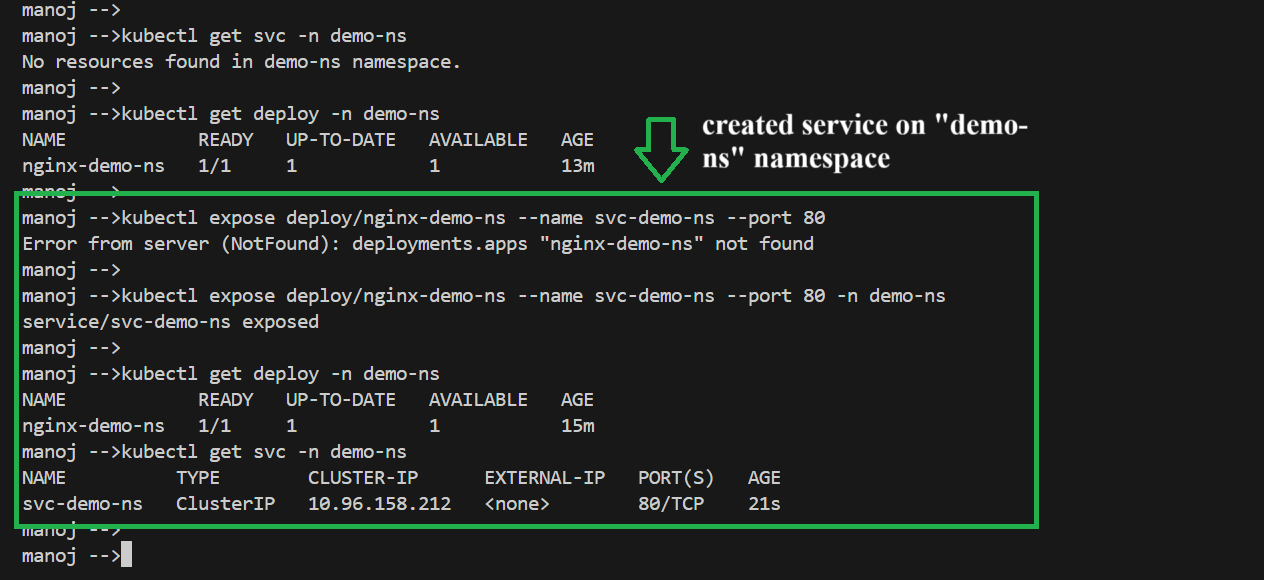
Now let me check <demo-namespace> can reach default namespace, using container ip





Now let me expose “service” in both deployment.





Now let me check can I access service present in default namespace from the “dev/qa namespace”



Using “fully qualified domain name” I can access the service present in different namespace.

