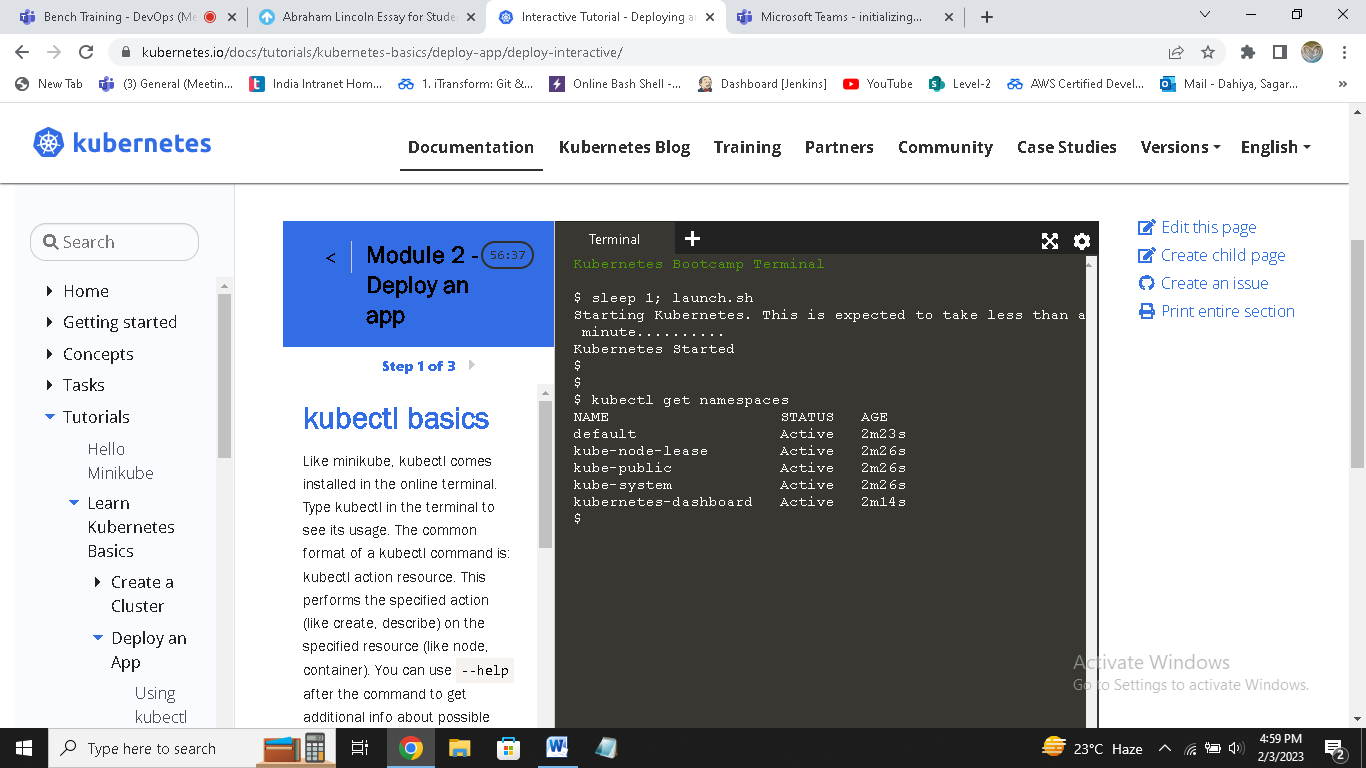
**KUBERNETES NAMESPACE**

**Namespace provides an additional qualification to a resource name. This is helpful when multiple teams are using the same cluster and there is a potential of name collision. It can be as a virtual wall between multiple clusters.**

**Functionality of Namespace**

**Following are some of the important functionalities of a Namespace in Kubernetes −**

* **Namespaces help pod-to-pod communication using the same namespace.**
* **Namespaces are virtual clusters that can sit on top of the same physical cluster.**
* **They provide logical separation between the teams and their environments.**



**Create a Namespace**

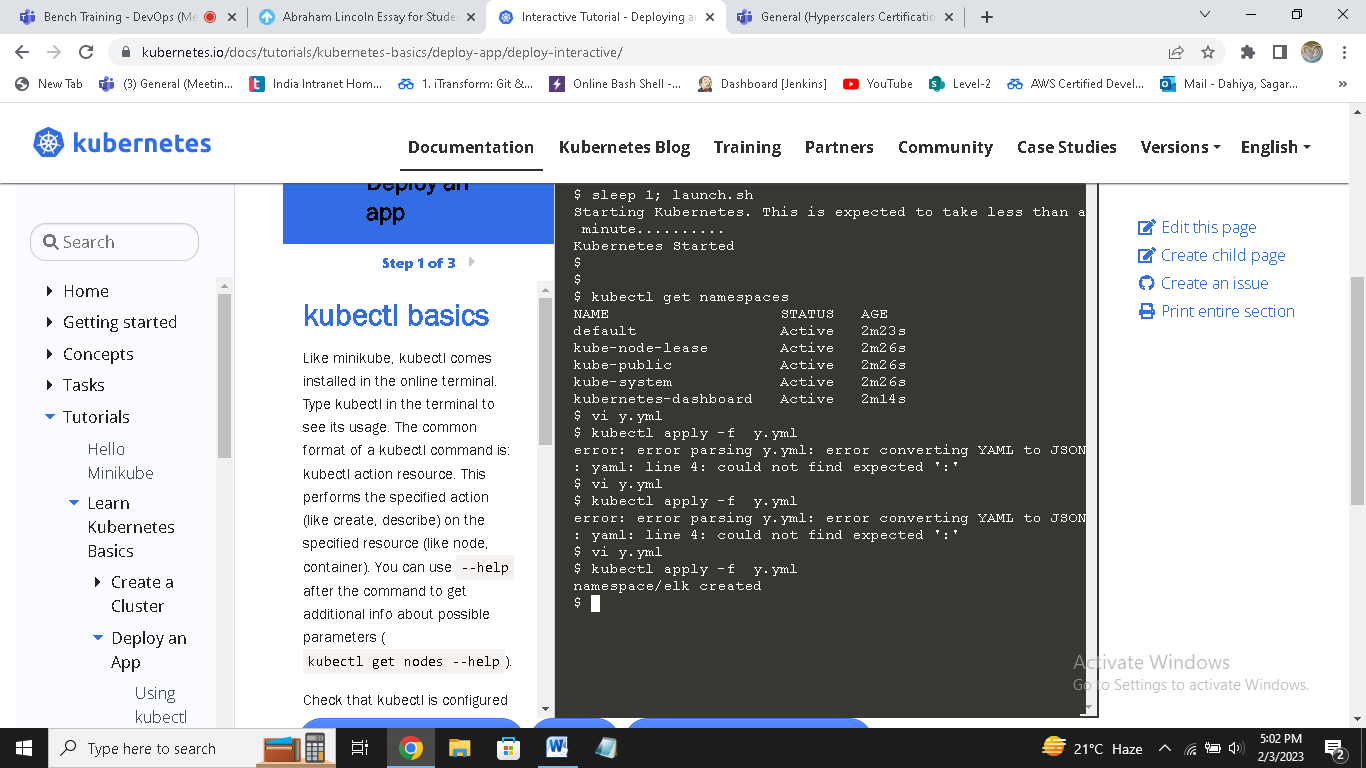
**The following command is used to create a namespace.**

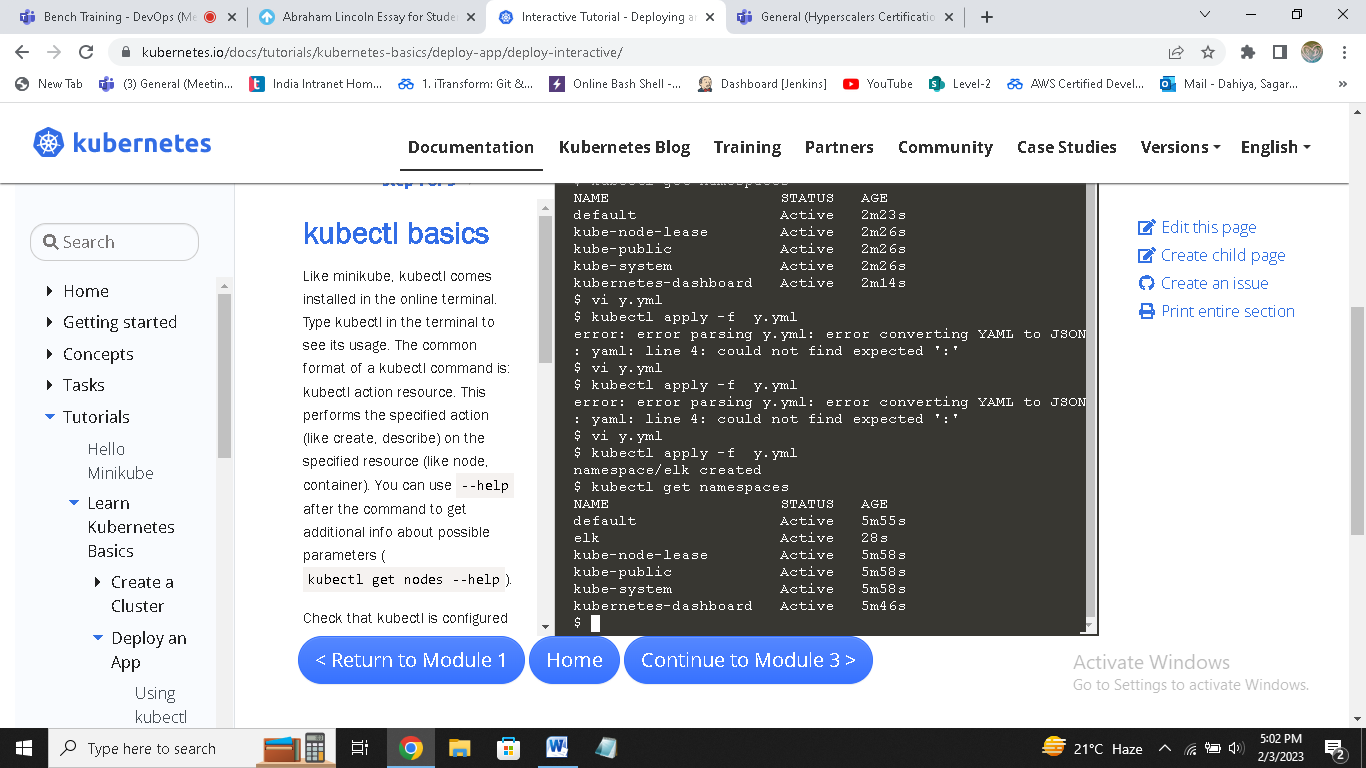
**apiVersion: v1**

**kind: Namespce**

**metadata**

**name: elk**





**Control the Namespace**

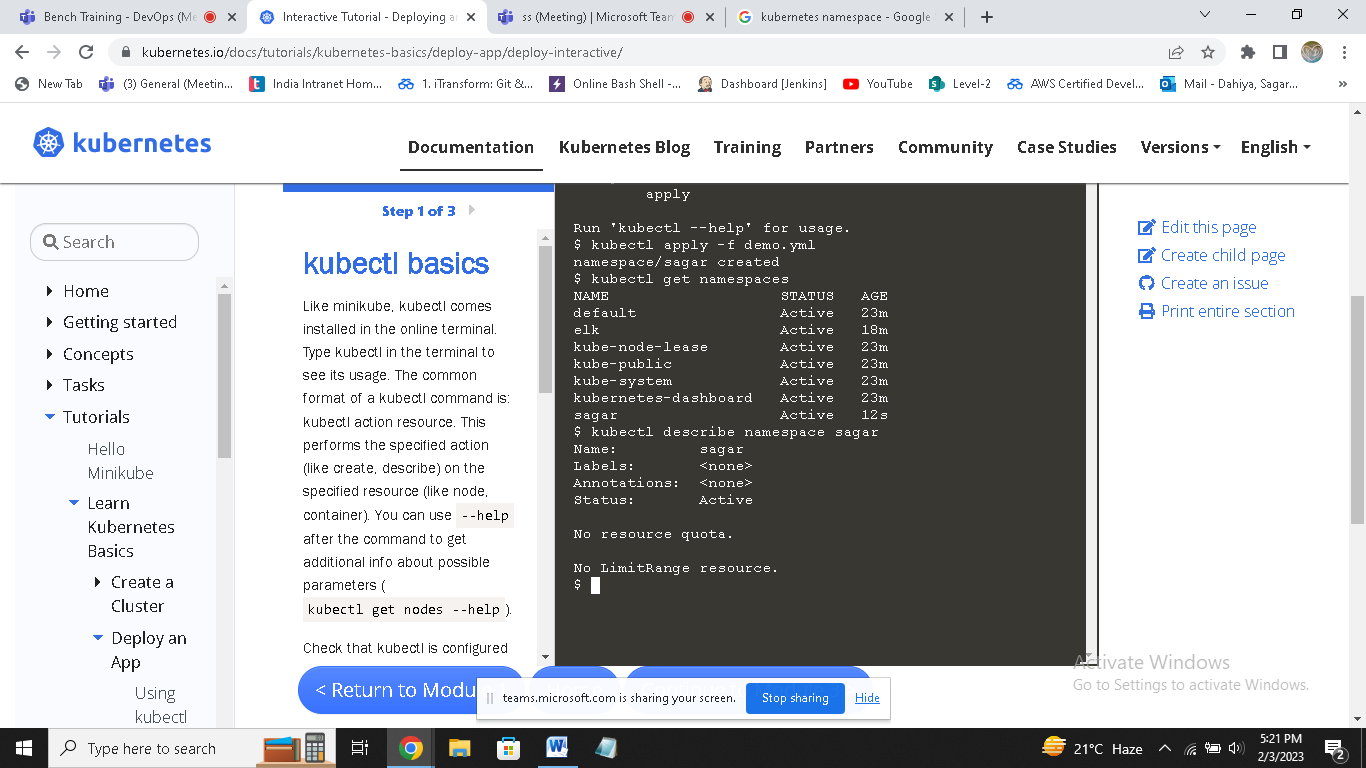
**The following command is used to control the namespace.**

**$ kubectl create –f namespace.yml ---------> 1**

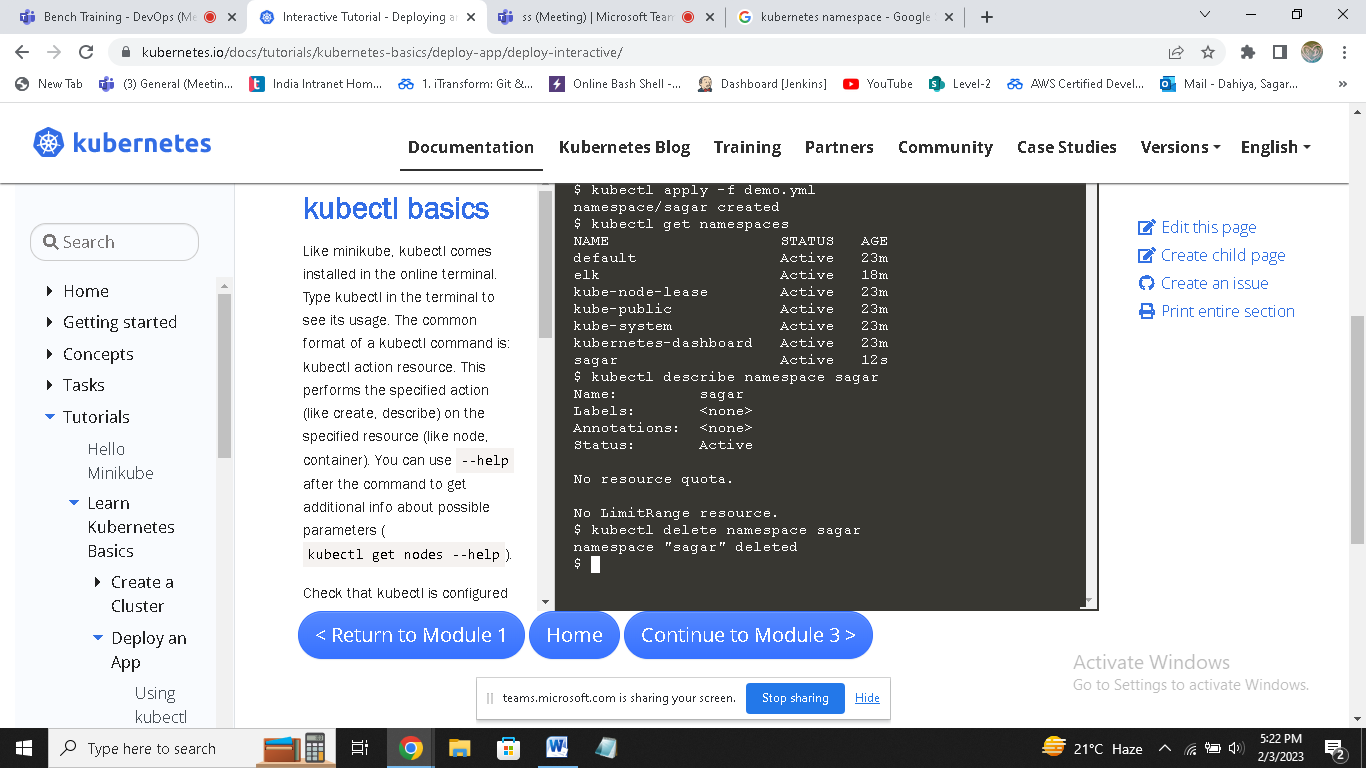
**$ kubectl get namespace -----------------> 2**

**$ kubectl get namespace <Namespace name> ------->3**

**$ kubectl describe namespace <Namespace name> ---->4**



**$ kubectl delete namespace <Namespace name>**



**In the above code,**

* **We are using the command to create a namespace.**
* **This will list all the available namespace.**
* **This will get a particular namespace whose name is specified in the command.**
* **This will describe the complete details about the service.**

==============================NAMESPACES===================================

apiVersion: v1

kind: Namespace

metadata:

name: dev

labels:

name: dev

=================================to create a pod=================

vi pod.yml

kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

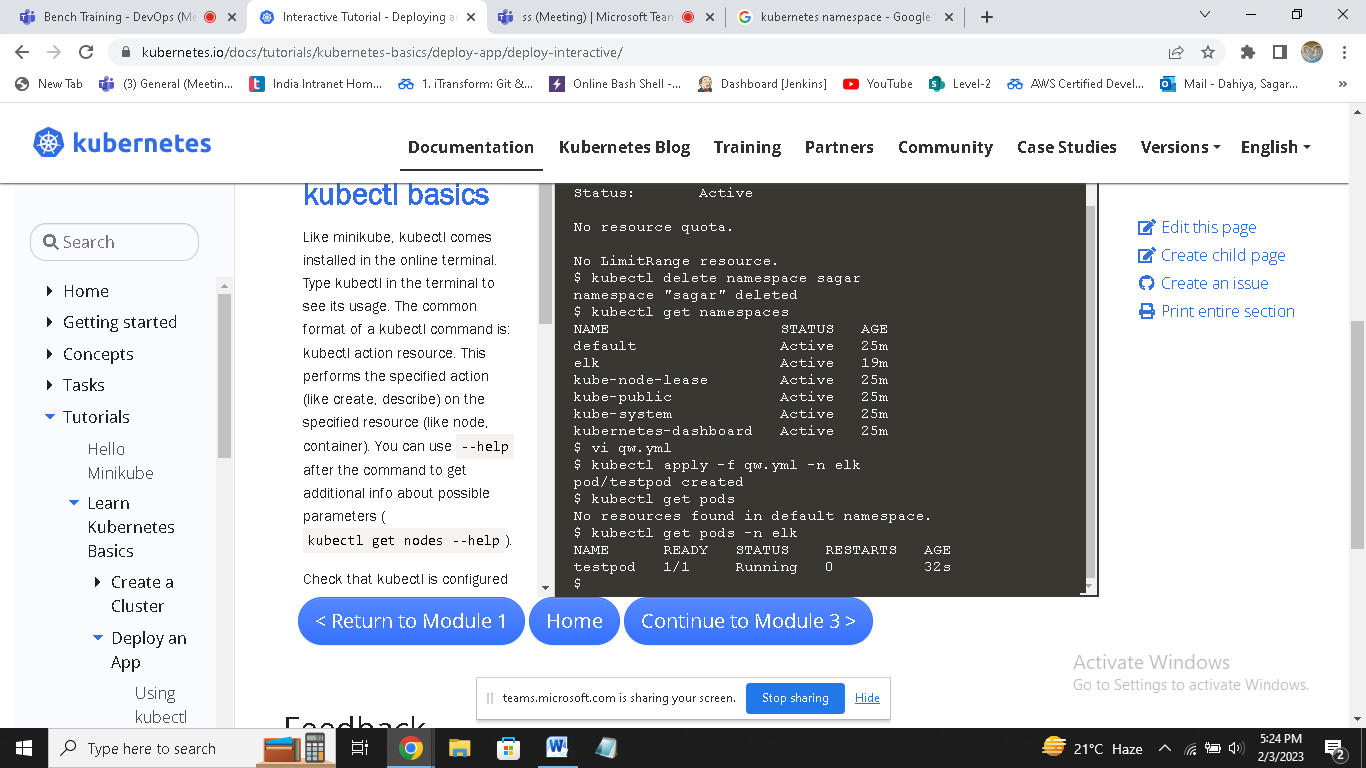
containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Technical Guftgu; sleep 5 ; done"]

restartPolicy: Never



**NODE**

A node is a working machine in Kubernetes cluster which is also known as a minion. They are working units which can be physical, VM, or a cloud instance.

**Each node has all the required configuration required to run a pod on it such as the proxy service and kubelet service along with the Docker, which is used to run the Docker containers on the pod created on the node.**

**They are not created by Kubernetes but they are created externally either by the cloud service provider or the Kubernetes cluster manager on physical or VM machines.**

**The key component of Kubernetes to handle multiple nodes is the controller manager, which runs multiple kind of controllers to manage nodes. To manage nodes, Kubernetes creates an object of kind node which will validate that the object which is created is a valid node.**

