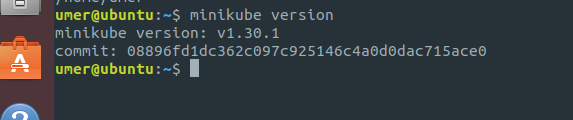
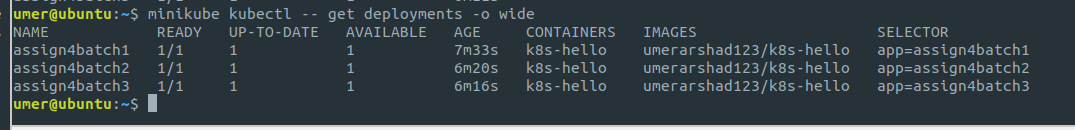
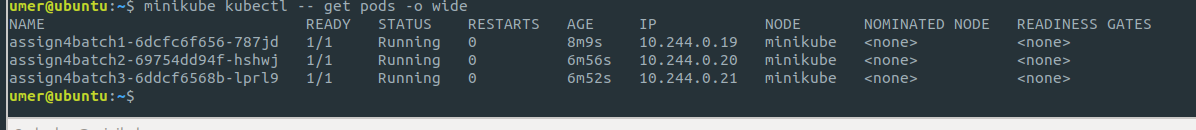
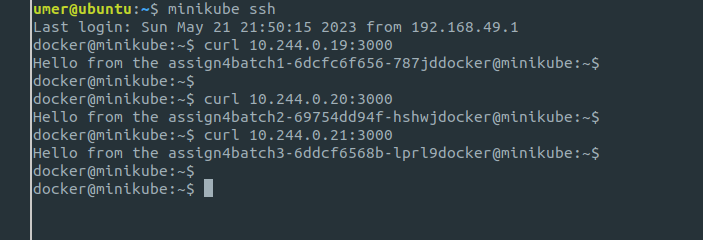
## Minikube Installation

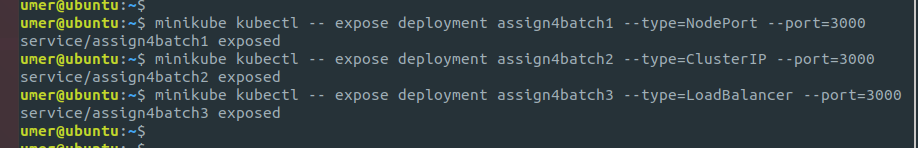
Installed minikube version 1.30.1 on Ubuntu VM

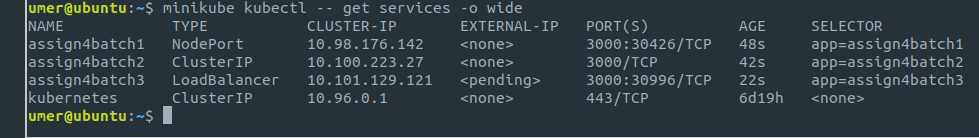


## Deploying Applications

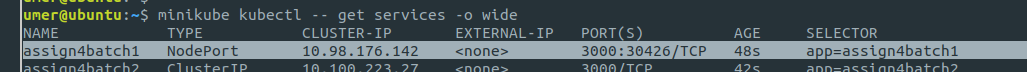
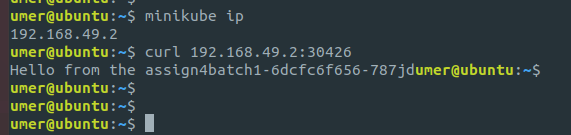
1. The custom docker image that displays the pod name pod name (or container id).   
   The image can be downloaded from: <https://hub.docker.com/repository/docker/umerarshad123/k8s-hello/general>
2. Created three deployments from this image each containing a single pod  
   minikube kubectl – create deployment assign4batch1 –image=umerarshad123/k8s-hello  
   minikube kubectl – create deployment assign4batch2 –image=umerarshad123/k8s-hello  
   minikube kubectl – create deployment assign4batch3 –image=umerarshad123/k8s-hello
3. All deployments are up and working as shown below:  
     
     
   

## Setting up Services





## Accessibility Demonstration

1. When a ‘clusterIP’ type service is created in kubernetes, the ip address assigned to the service is only accessible from within the cluster. The primary reason of a clusterIP service is to provide internal communication and load balancing among pods within the cluster. This is why pods are not accessible outside the cluster in clusterIP service.
2. The NodePort service is exposing the a port to access the service using the node’s ip  
     
   Now we can do a curl request to the node’s ip on port 30426 to access the underlying pod  
   since we are using minikube node’s ip is same as minikube ip in this case:  
   
3. In my VM I am using minikube so the service type LoadBalancer is simulated using a NodePort service.  
   