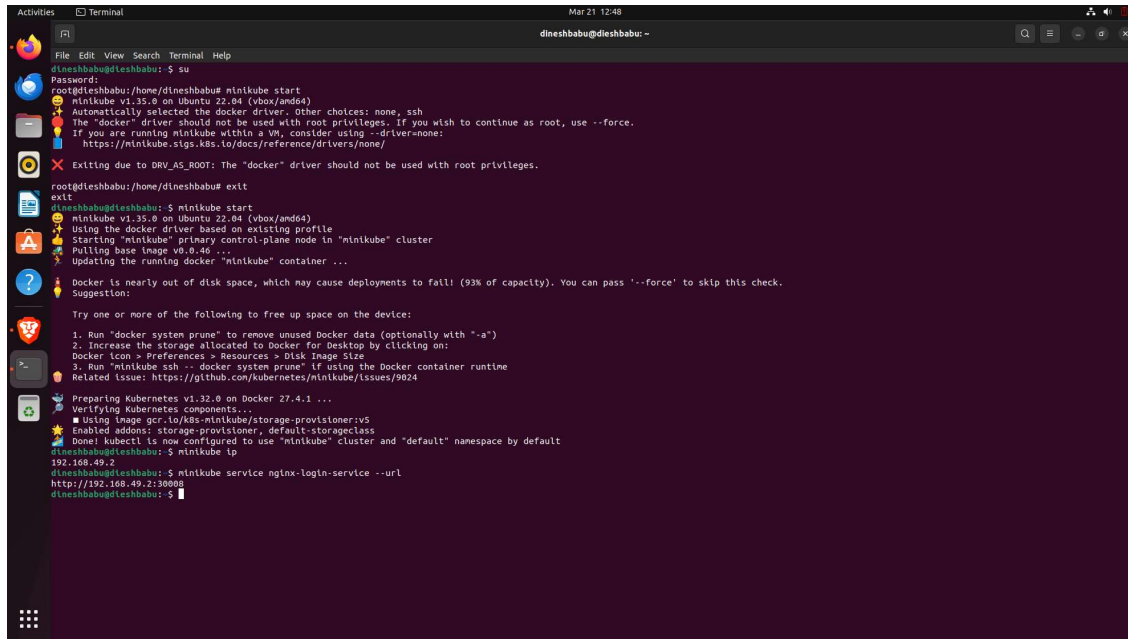


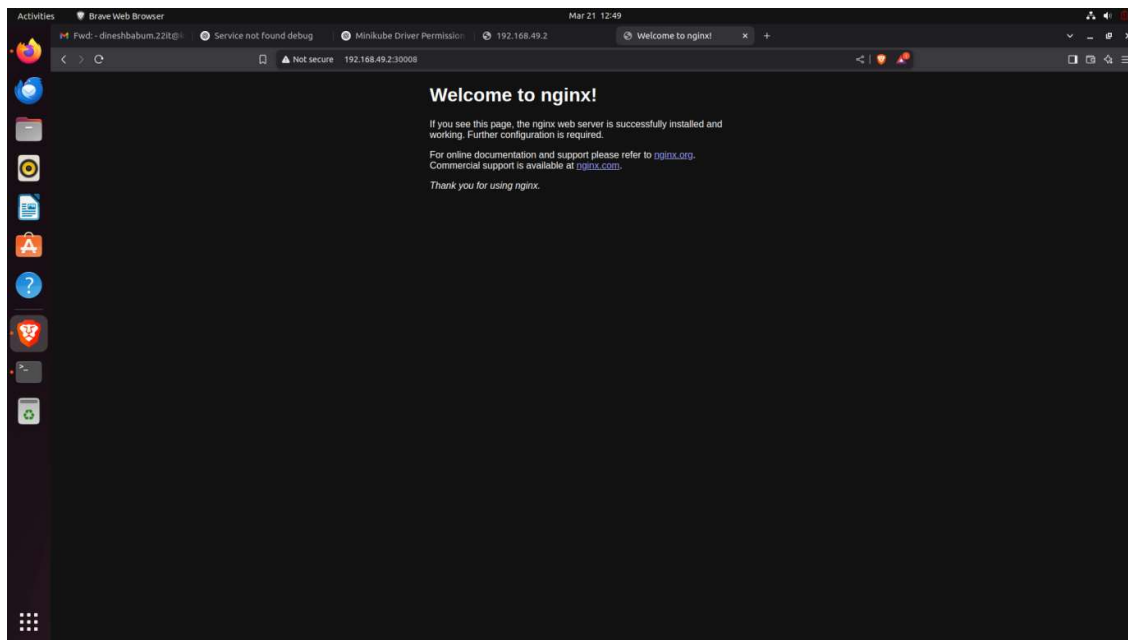
TASK-3:

MINIKUBE AND KUBECTL VERSION



```
dineshbabu@dieshbabu:~$ su
root@dieshbabu:/home/dineshbabu# minikube start
minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
* Automatically selected the docker driver. Other choices: none, ssh
* The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
* If you are running minikube within a VM, consider using --driver=none:
  https://minikube.sigs.k8s.io/docs/reference/drivers/none/
* Exiting due to DRV_AS_ROOT: The "docker" driver should not be used with root privileges.
root@dieshbabu:/home/dineshbabu# exit
exit
dineshbabu@dieshbabu:~$ minikube start
minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Updating the running docker "minikube" container ...
* Docker is nearly out of disk space, which may cause deployments to fail! (93% of capacity). You can pass '--force' to skip this check.
  Suggestion:
  Try one or more of the following to free up space on the device:
  1. Run "docker system prune" to remove unused Docker data (optionally with "-a")
  2. Increase the storage allocated to Docker for Desktop by clicking on:
     Docker icon > Preferences > Resources > Disk Image Size
  3. Run "minikube ssh -- docker system prune" if using the Docker container runtime
  Related issue: https://github.com/kubernetes/minikube/issues/9924
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
* Verifying Kubernetes components...
* Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
dineshbabu@dieshbabu:~$ minikube ip
192.168.49.2
dineshbabu@dieshbabu:~$ minikube service nginx-login-service --url
http://192.168.49.2:30008
dineshbabu@dieshbabu:~$
```

OUTPUT:



```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-login
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-login
  template:
    metadata:
      labels:
        app: nginx-login
    spec:
      containers:
        - name: nginx-login
          image: nginx:latest
          ports:
            - containerPort: 80
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-login-service
spec:
  type: NodePort
  selector:
    app: nginx-login
  ports:
```

- protocol: TCP
- port: 80
- targetPort: 80
- nodePort: 30008