Lesson 7 Lesson-End Project

**Deploy WordPress and MySQL Using Persistent Volume**

**Project agenda:** Deploy WordPress and MySQL using persistent volume

**Description:** Your organization needs WordPress and MySQL applications, where WordPress must be deployed using host path and MySQL must be deployed using NFS

**Tools required:** Kubeadm, kubectl, kubelet, and docker

**Prerequisites:** Kubeadm, kubectl, kubelet, and docker must be installed

**Expected deliverables:** A Kubernetes cluster with high availability enabled

**Steps to be followed:**

1. Create a cluster
2. Create an NFS server
3. Deploy NFS client on worker nodes
4. Create a MySQL manifest file and deploy it using NFS-based persistent volume
5. Create a WordPress manifest file and deploy it using host-path-based persistent volume

**Step** **1**: **Create the Kubernetes cluster**

|  |
| --- |
| **Note:** Refer Demo 1 of Lesson 4 to create a Kubernetes cluster |

**Step 2: Create an NFS server**

2.1 Identify any one server to deploy an NFS server on top of, and create a directory on any machine that you wish to share with the client system

**sudo su**

**mkdir -p /data**

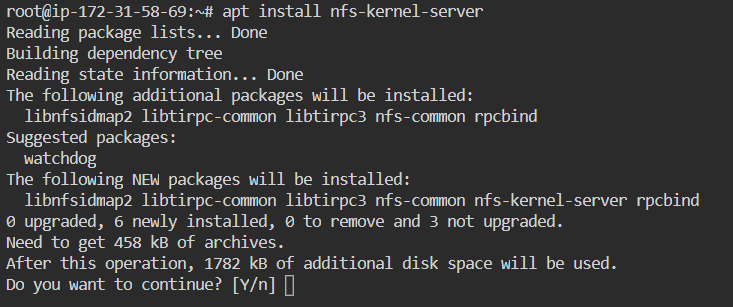
**ls -alrt /data/**

***Text

Description automatically generated***

2.2Run the following command to install the NFS kernel server on the machine:

**sudo apt install nfs-kernel-server**

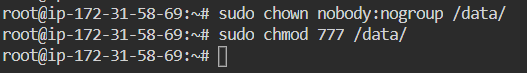
******

2.3Change the owner user and group to **nobody** and **nogroup**

**sudo chown nobody:nogroup /data/**

Set permissions to **777** to allow everyone to read, write, and execute files in this directory

***sudo chmod 777 /data/***



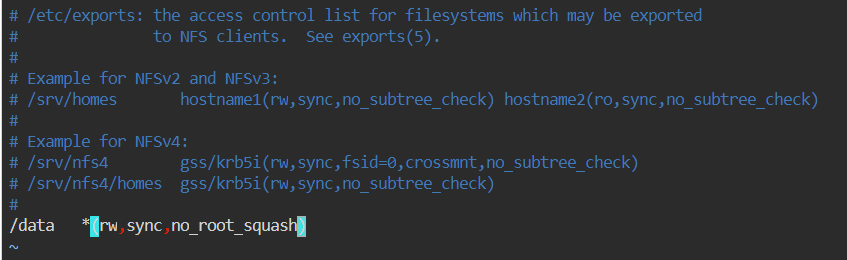
2.4Open the exports file in the **/etc** directory for permission to access the host server machine

**sudo vi /etc/exports**

Add the following code to the file:

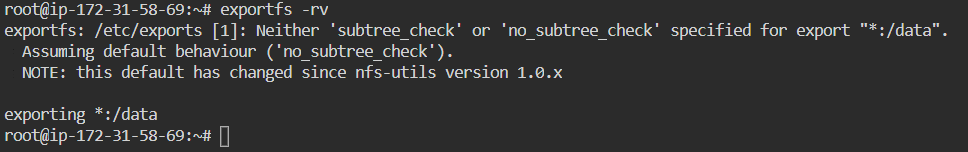
**/data \*(rw,sync,no\_root\_squash)**

Note: Exit the file and save the changes



Use the **exportfs** command to export all shared folders you registered in the **/etc/exports** file after making the appropriate changes

**sudo exportfs -rv**



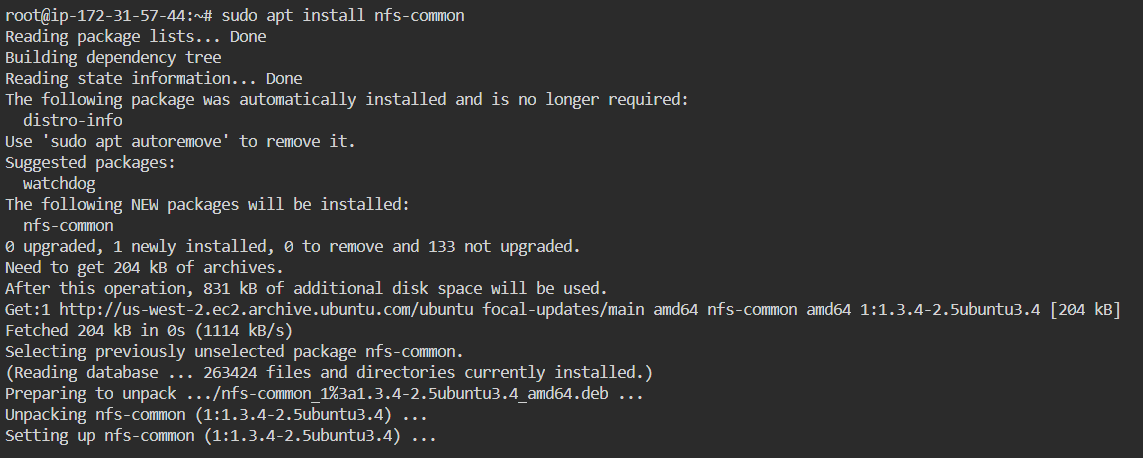
2.5 Restart the NFS kernel server to apply the configuration changes

**sudo systemctl restart nfs-kernel-server**

**Step 3: Deploy NFS client on worker nodes**

3.1 Run the command given below to deploy the NFS client on all worker nodes as part of the cluster:

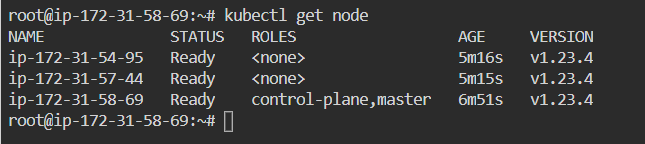
**sudo apt install nfs-common**



**Step 4: Create a MySQL manifest file and deploy it using NFS-based persistent volume**

4.1 Validate connectivity between Kubernetes master and worker nodes using the below command:

***kubectl get node***



4.2 Create the MySQL manifest file with **nfs-mysql.yaml** using the below content:

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: mysql-nfs**

**spec:**

**capacity:**

**storage: 1Gi**

**accessModes:**

**- ReadWriteMany**

**persistentVolumeReclaimPolicy: Recycle**

**mountOptions:**

**- hard**

**- nfsvers=4.1**

**nfs:**

**path: /data**

**server: 172.31.58.69**

**---**

**kind: PersistentVolumeClaim**

**apiVersion: v1**

**metadata:**

**name: mysql-nfs**

**spec:**

**accessModes:**

**- ReadWriteMany**

**resources:**

**requests:**

**storage: 500Mi**

**---**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: mysql**

**labels:**

**app: mysql-wordpress**

**spec:**

**ports:**

**- port: 3306**

**selector:**

**app: mysql-wordpress**

**product: mysql**

**---**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mysql**

**labels:**

**app: mysql-wordpress**

**spec:**

**selector:**

**matchLabels:**

**app: mysql-wordpress**

**product: mysql**

**strategy:**

**type: Recreate**

**template:**

**metadata:**

**labels:**

**app: mysql-wordpress**

**product: mysql**

**spec:**

**containers:**

**- image: mysql**

**name: mysql-container**

**env:**

**- name: MYSQL\_DATABASE**

**value: wordpress**

**- name: MYSQL\_ROOT\_PASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mysql-secret-password**

**key: password**

**ports:**

**- containerPort: 3306**

**name: mysql**

**volumeMounts:**

**- name: mysql-storage**

**mountPath: /var/lib/mysql**

**volumes:**

**- name: mysql-storage**

**persistentVolumeClaim:**

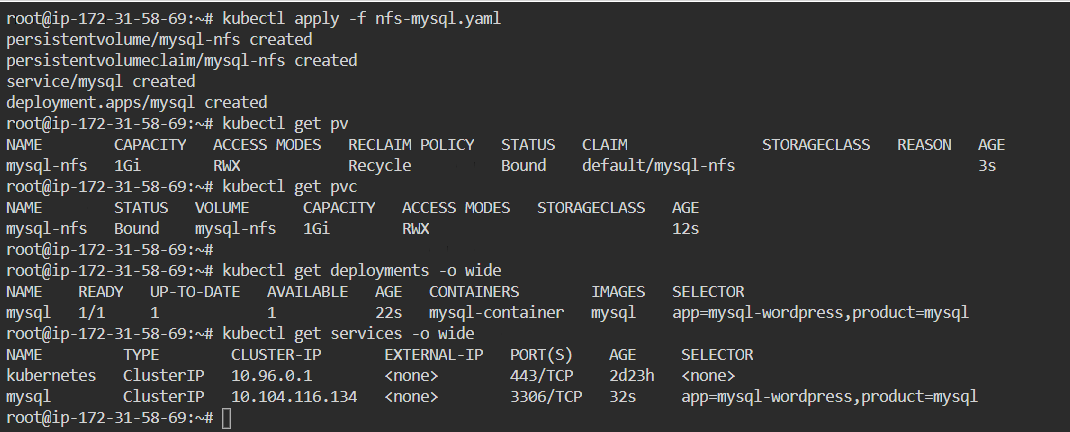
**claimName: mysql-nfs**

**Note:** The below section may change if you have a different NFS server and share folder available:

**nfs:**

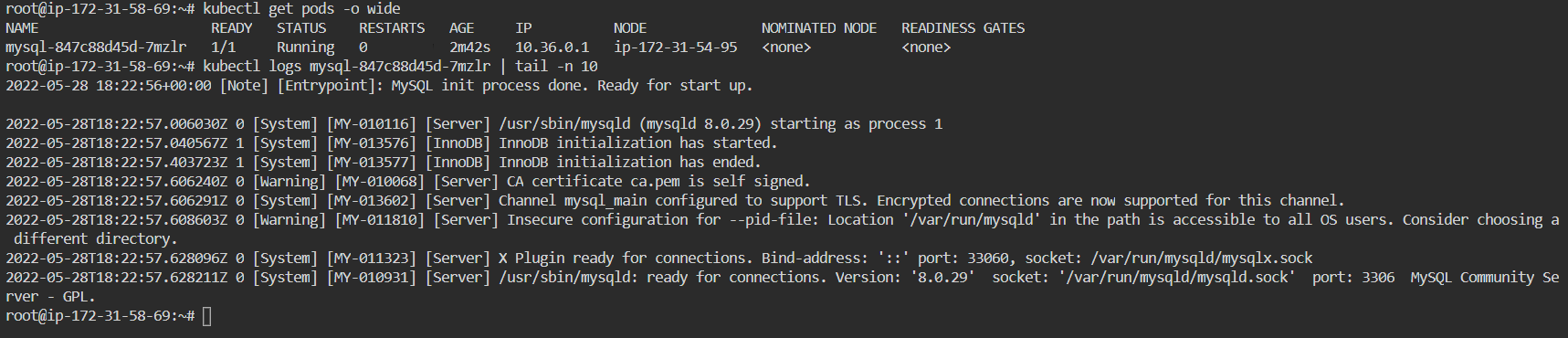
**path: /data**

**server: 172.31.58.69**

****

4.3Once the Deployment is hosted on the Kubernetes cluster, validate that MySQL is up and running and that there aren’t any issues in Pod logs

**kubectl logs <pod\_name>**



**Step 5: Create a WordPress manifest file and deploy it using host-path-based persistent volume**

5.1 Before deploying WordPress, create a host-path-based persistent volume to make the data inside the WordPress image persistent

5.2 Use the below manifest YAML content to create PV and PVC

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: hostpath-pv**

**labels:**

**type: hostpath**

**spec:**

**capacity:**

**storage: 2Gi**

**accessModes:**

**- ReadWriteMany**

**storageClassName: ""**

**persistentVolumeReclaimPolicy: Delete**

**hostPath:**

**type: DirectoryOrCreate**

**path: "/opt/wordpress"**

**---**

**kind: PersistentVolumeClaim**

**apiVersion: v1**

**metadata:**

**name: wordpress-hostpath**

**spec:**

**accessModes:**

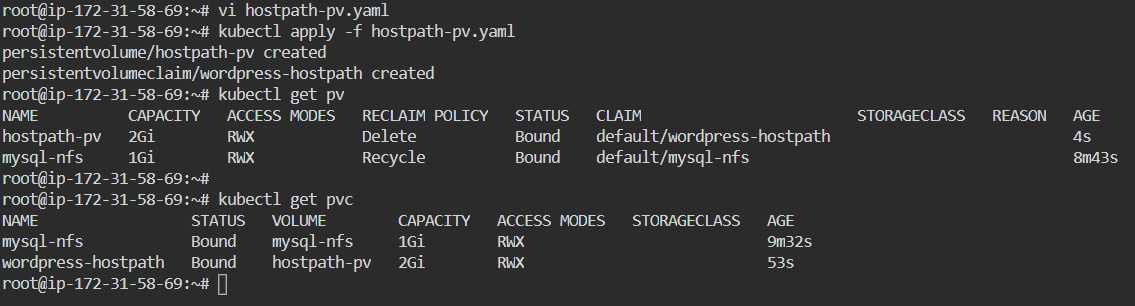
**- ReadWriteMany**

**storageClassName: ""**

**resources:**

**requests:**

**storage: 500Mi**



5.3 Use the below manifest YAML content to create a WordPress deployment and services

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: wordpress**

**labels:**

**app: mysql-wordpress**

**spec:**

**ports:**

**- port: 80**

**selector:**

**app: mysql-wordpress**

**tier: frontend**

**type: NodePort**

**---**

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: wordpress**

**labels:**

**app: mysql-wordpress**

**spec:**

**selector:**

**matchLabels:**

**app: mysql-wordpress**

**tier: frontend**

**strategy:**

**type: Recreate**

**template:**

**metadata:**

**labels:**

**app: mysql-wordpress**

**tier: frontend**

**spec:**

**containers:**

**- image: wordpress**

**name: wordpress**

**env:**

**- name: WORDPRESS\_DB\_HOST**

**value: mysql**

**- name: WORDPRESS\_DB\_USER**

**value: root**

**- name: WORDPRESS\_DB\_PASSWORD**

**valueFrom:**

**secretKeyRef:**

**name: mysql-secret-password**

**key: password**

**ports:**

**- containerPort: 80**

**name: wordpress**

**volumeMounts:**

**- name: wordpress-storage**

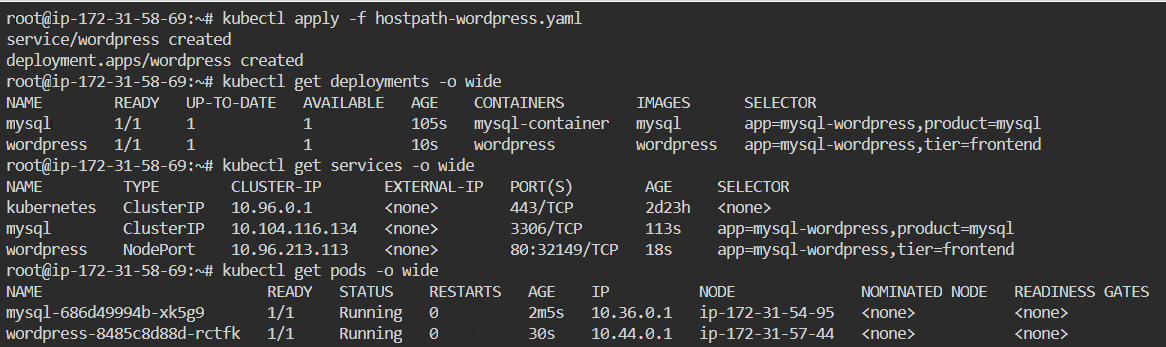
**mountPath: /var/www/html**

**volumes:**

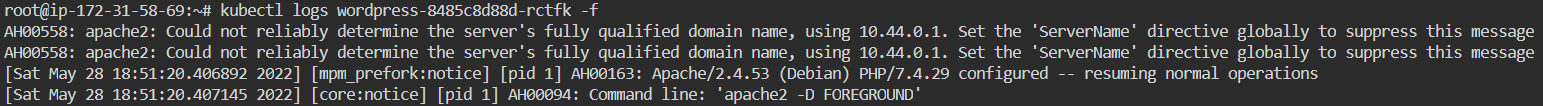
**- name: wordpress-storage**

**persistentVolumeClaim:**

**claimName: wordpress-hostpath**



5.4 Once deployed, validate WordPress Pods to ensure it is working as expected:



Then, open the Firefox browser and put the below URL to access the WordPress application

http://localhost:<node\_port>

Provide website-related information and proceed with installation:

