

## СОЗДАНИЕ PV С ИСПОЛЬЗОВАНИЕМ NFS.

### Persistent Volume (PV):

PV – это объект в Kubernetes, который представляет собой абстракцию над реальным хранилищем (например, диск, файловая система, облачное хранилище и т. д.). PV создается администратором кластера и может быть использован для хранения данных, которые должны быть доступны даже после перезапуска подов.

### Persistent Volume Claim (PVC):

PVC – это запрос на хранилище, который делает приложение (или пользователь). PVC указывает, сколько места нужно для хранилища, какие права доступа требуются и другие параметры. Kubernetes автоматически связывает PVC с подходящим PV, если такой существует, или создает новый PV, если это предусмотрено политиками.

### Шаг 1: Разверните NFS сервер с готового образа из панели управления Ubuntu 22.04 (NFS-IMAGE).

В образе по умолчанию расширена папка `/mnt/nfs_share`.

Убедитесь, что NFS образ находится в одной локальной подсети с кластером Kubernetes.

### Шаг 2: Создайте Persistent Volume (PV) в Kubernetes

- 1) Создайте объект Persistent Volume в Kubernetes, который будет использовать NFS как источник хранилища, затем создайте манифест для PV: путем создания YAML файла, например `nfs-pv.yaml`:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-pv
spec:
  capacity:
    storage: 10Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  storageClassName: nfs-storage
  nfs:
    path: /mnt/nfs_share
    server: <IP_адрес_вашего_NFS_сервера>
```

Замените `<IP_адрес_вашего_NFS_сервера>` на IP-адрес вашего NFS сервера.

- 2) Примените манифест в Kubernetes:

```
kubectl apply -f nfs-pv.yaml
```

- 3) Проверьте, что PV создан:

```
kubectl get pv
```

### Шаг 3: Создайте Persistent Volume Claim (PVC)

- 1) Создайте PVC, чтобы приложение в Kubernetes могло запросить доступ к этому хранилищу затем создайте манифест для PVC: путем создания файла, например `nfs-pvc.yaml`:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: nfs-storage
```

- 2) Примените PVC в Kubernetes:

```
kubectl apply -f nfs-pvc.yaml
```

- 2) Проверьте, что PVC создан:

```
kubectl get pvc
```

### Шаг 4: Использование PVC в подах

- 1) Теперь, когда у вас есть PVC, вы можете использовать его в подах. Пример манифеста для пода с монтированным NFS:

```
apiVersion: v1
kind: Pod
metadata:
  name: nfs-pod
spec:
  containers:
    - name: nginx
      image: nginx
      volumeMounts:
        - mountPath: /usr/share/nginx/html
          name: nfs-volume
  volumes:
    - name: nfs-volume
      persistentVolumeClaim:
        claimName: nfs-pvc
```

Примените манифест пода:

```
kubectl apply -f nfs-pod.yaml
```

Расширение хранилища в Kubernetes зависит от того, как настроен ваш NFS-сервер и какие Persistent Volume (PV) и Persistent Volume Claim (PVC) вы используете. Вот пошаговый процесс для увеличения доступного места:

## РАСШИРЕНИЕ ХРАНИЛИЩА НА СТОРОНЕ NFS-СЕРВЕРА

Если у вас закончилось пространство на уровне NFS-сервера, вам нужно:

- 1) Добавить больше места к экспортируемой директории:

Если директория `/mnt/nfs_share` находится на отдельном диске, увеличьте размер этого диска с панели управления (при выборе варианта с перезагрузкой диск расширится автоматически).

- 2) Убедитесь, что размер доступного пространства обновлен: После добавления места убедитесь, что NFS-сервер видит новое доступное пространство. Например: `df -h /mnt/nfs_share`
- 3) Нет необходимости перезапускать NFS-сервер, если вы просто увеличили доступное место.

## ОБНОВЛЕНИЕ PV В KUBERNETES

После того, как на стороне NFS-сервера добавлено место, обновите размер Persistent Volume (PV) в Kubernetes. PV – это статический объект, и его нужно вручную изменить.

- 1) Обновите манифест PV: Найдите ваш PV (например, `nfs-pv`) и увеличьте значение в поле `capacity.storage`. Например:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-pv
spec:
  capacity:
    storage: 20Gi # Увеличено с 10Gi до 20Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  storageClassName: nfs-storage
  nfs:
    path: /mnt/nfs_share
    server: <IP_адрес_вашего_NFS_сервера>
```

- 2) Примените изменения:

```
kubectl apply -f nfs-pv.yaml
```

- 3) Проверьте обновленный размер PV:

```
kubectl get pv
```

## ОБНОВЛЕНИЕ PVC В KUBERNETES.

Если PVC связан с PV, вы также можете обновить его размер. Для этого:

- 1) Измените манифест PVC: Увеличьте значение в `resources.requests.storage`. Например:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 20Gi # Увеличено с 10Gi до 20Gi
  storageClassName: nfs-storage
```

- 2) Примените изменения:

```
kubectl apply -f nfs-pvc.yaml
```

- 3) Проверьте статус PVC и убедитесь, что PVC был обновлен:

```
kubectl get pvc
```

## ОБНОВЛЕНИЕ ПРИЛОЖЕНИЯ

Если ваше приложение использует под, который монтирует PVC, обычно никаких дополнительных действий не требуется. Объем хранилища будет увеличен автоматически. Однако убедитесь, что ваше приложение может правильно распознать новые размеры тома.