---

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv-nautilus

spec:

capacity:

storage: 5Gi

accessModes:

- ReadWriteOnce

storageClassName: manual

hostPath:

path: /mnt/data

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pvc-nautilus

spec:

accessModes:

- ReadWriteOnce

storageClassName: manual

resources:

requests:

storage: 3Gi

---

apiVersion: v1

kind: Pod

metadata:

name: pod-nautilus

spec:

volumes:

- name: storage-nautilus

persistentVolumeClaim:

claimName: pvc-nautilus

containers:

- name: container-nautilus

image: httpd:latest

ports:

- containerPort: 80

volumeMounts:

- name: storage-nautilus

mountPath: /usr/local/apache2/htdocs

---

apiVersion: v1

kind: Service

metadata:

name: web-nautilus

spec:

type: NodePort

ports:

- port: 80

nodePort: 30008

name: http

selector:

name: httpd-app

Hi. I think the default document root of httpd:latest image is /usr/local/apache2/htdocs/ as per the document in https://hub.docker.com/\_/httpd. Regarding the web site access check using curl command, how about executing from the jump\_host instead of within the container? The service should be open at port 30008 on the node, and the node ip address could be checked by a command like kubectl get pods -o wide.

---

apiVersion: v1

kind: PersistentVolume

metadata:

name: pv-xfusion

spec:

capacity:

storage: 4Gi

accessModes:

- ReadWriteOnce

storageClassName: manual

hostPath:

path: /mnt/dba

---

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pvc-xfusion

spec:

accessModes:

- ReadWriteOnce

storageClassName: manual

resources:

requests:

storage: 3Gi

---

apiVersion: v1

kind: Pod

metadata:

name: pod-xfusion

spec:

volumes:

- name: storage-xfusion

persistentVolumeClaim:

claimName: pvc-xfusion

containers:

- name: container-xfusion

image: nginx:latest

ports:

- containerPort: 80

volumeMounts:

- name: storage-xfusion

mountPath: /usr/share/nginx/html

---

apiVersion: v1

kind: Service

metadata:

name: web-xfusion

spec:

type: NodePort

selector:

app: nginx-app

type: front-end

ports:

- port: 80

targetPort: 80

nodePort: 30008

**NEWSSSSSSSSSSSSSSSSSSSSSSS**

**---**

**apiVersion: v1**

**kind: PersistentVolume**

**metadata:**

**name: pv-devops**

**spec:**

**capacity:**

**storage: 5Gi**

**accessModes:**

**- ReadWriteOnce**

**storageClassName: manual**

**hostPath:**

**path: /mnt/devops**

**---**

**apiVersion: v1**

**kind: PersistentVolumeClaim**

**metadata:**

**name: pvc-devops**

**spec:**

**accessModes:**

**- ReadWriteOnce**

**storageClassName: manual**

**resources:**

**requests:**

**storage: 1Gi**

**---**

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: pod-devops**

**spec:**

**volumes:**

**- name: storage-devops**

**persistentVolumeClaim:**

**persistentVolumeClaim:**

**claimName: pvc-devops**

**containers:**

**- name: container-devops**

**image: httpd:latest**

**ports:**

**- containerPort: 80**

**volumeMounts:**

**- name: storage-devops**

**mountPath: /usr/local/apache2/**

**---**

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: web-devops**

**spec:**

**type: NodePort**

**ports:**

**- port: 80**

**nodePort: 30008**

**name: http**

**selector:**

**name: pod-devops**