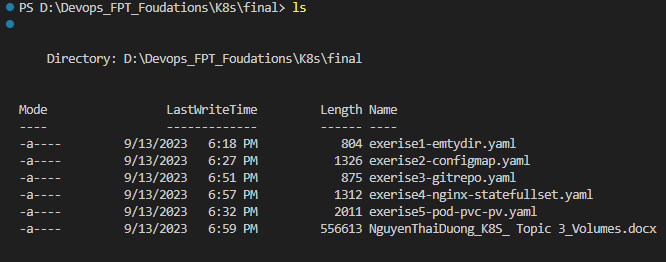
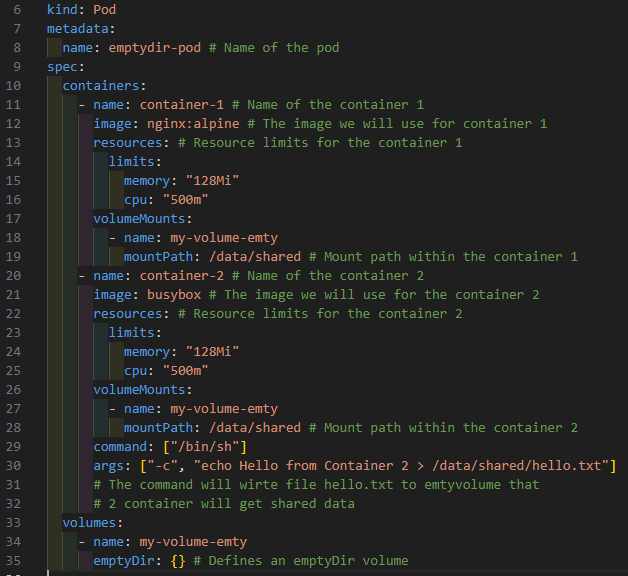
K8s Volumes09 – Nguyễn Thái Dương

Tất cả file yaml được viết tương ứng với mỗi bài tập lap

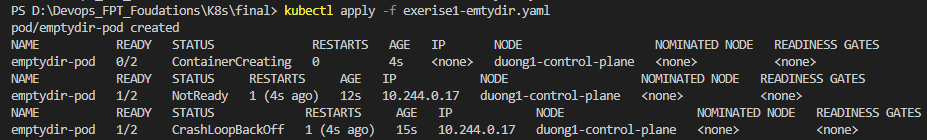


1 Create a nginx app with emptyDir volume (1.5)

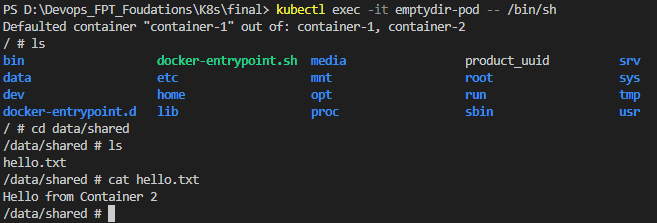
Chúng ta tạo một file yaml trong đó có khởi tạo config 1 pod, 2 container được gắn với volume my-volume-emty, volume này sẽ được định nghĩa là emtyDir ( volume này chia sẻ dữ liệu giữa các container cùng pod ) sau đó chúng ta viết một file trong đường dẫn volume dùng chung để check



Chúng ta apply file này và check xem đã có pod chạy chưa



Pod đã chạy, vì config như trên thì ta đoán được conttaner 2 busy box đã chết nhưng nó có ghi được file cùng shared cho container 1 nginx không ta sẽ vào check



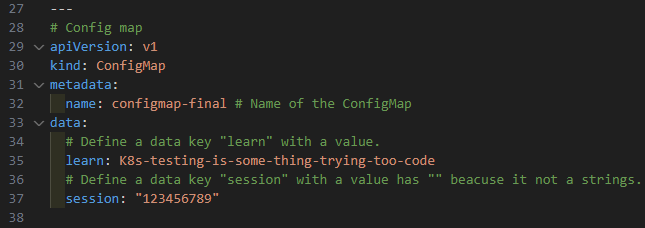
Theo như hình ảnh thì ta đã vào được pod và được mặc định vào container 1 nginx, chúng ta check file ở đường dẫn cùng shared volume thì thấy thằng busybox lúc khởi tạo đã ghi được file vào đường dẫn cùng shared sử dụng emtydir volume mount

Nếu pod bị mất thì dữ liệu trong emty volume mà các container cùng được shared cũng mất

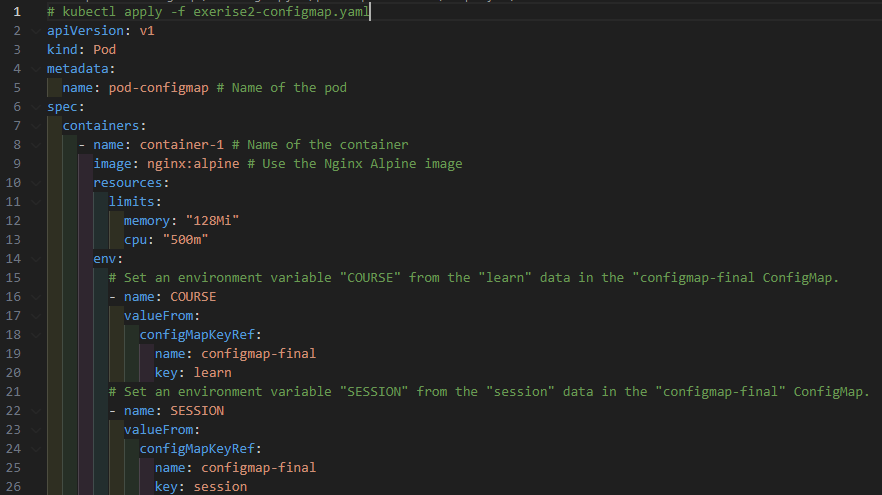


2 Create a nginx app with Config map volume (1.5)

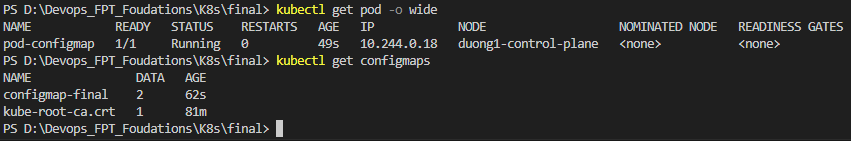
Chúng ta viết file khởi tạo file yaml cho configmap có 2 biến learn và session với giá trị tương ứng của nó



Chúng ta sau đó viết file yaml config 1 pod tên “pod -configmap” với một container dùng nginx tên “container-1” ta có truyền vào container đó các biến đã tạo ở file configmap với biến learn trong container sẽ được đặt là biến môi tường COURSE và session được truyền vào container là SESSION

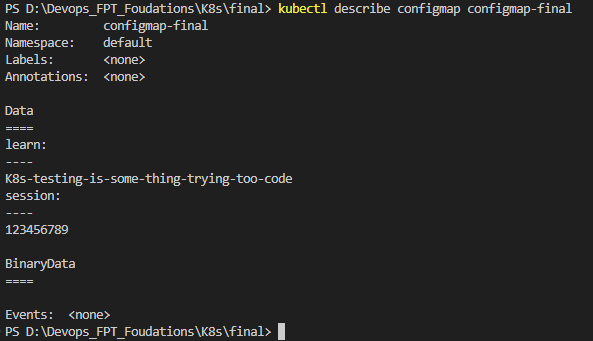


Chúng ta apply và check pod cùng với configmap

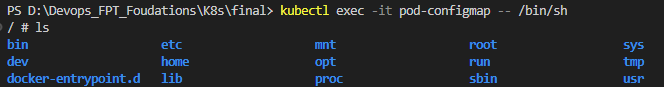


Ta thấy configmap-final đã được tạo

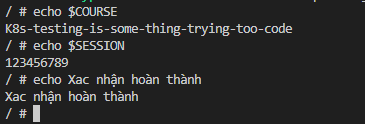
Ta mô tả nó

xác nhận các biến trong yaml đã có

Ta vào container 1 nginx trong pod ta mới tạo

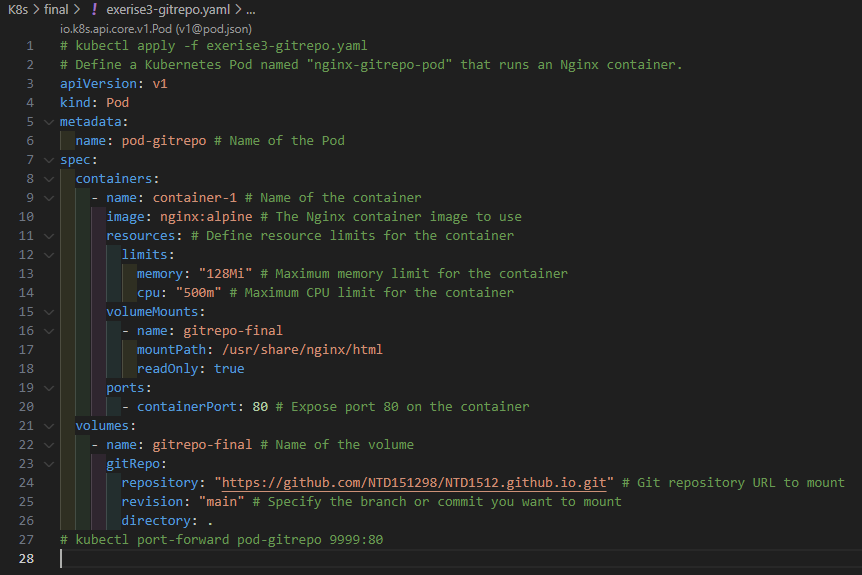


Ta check các biến đã vào container chưa

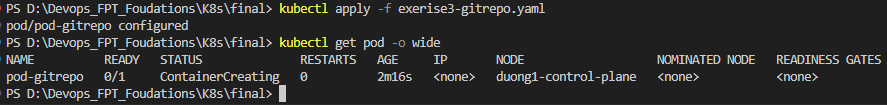
Vậy các biến COURSE và SESSION đã nhận giá trị từ volume configmap-final

3 Create a nginx app with gitRepo volume (1)

Chúng ta viết file yaml để tạo gitrepo volume với nginx pod, gitRepo volume nó sẽ clone git repo của chúng ta vào đường dẫn volume trên container với mountPath: /usr/share/nginx/html

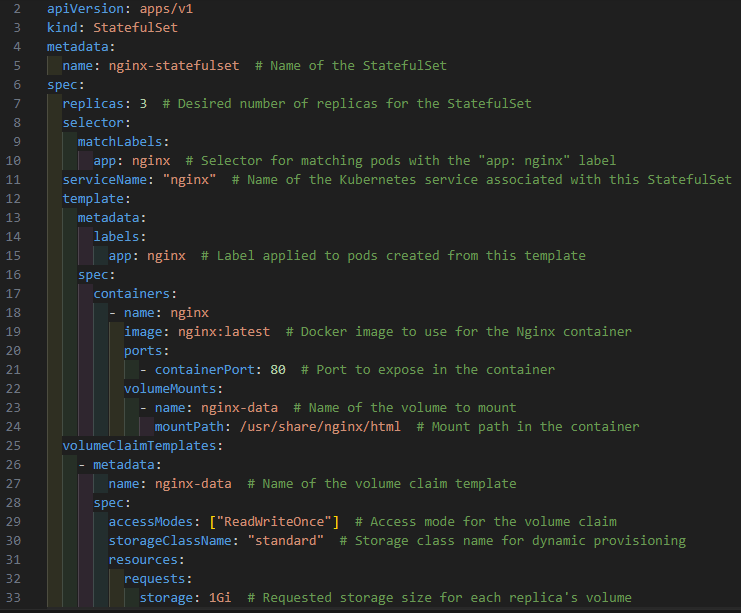


Ta apply và đợi container khởi tạo, theo việc gán vào gitRepo nó sẽ clone đường dẫn git URL in yaml

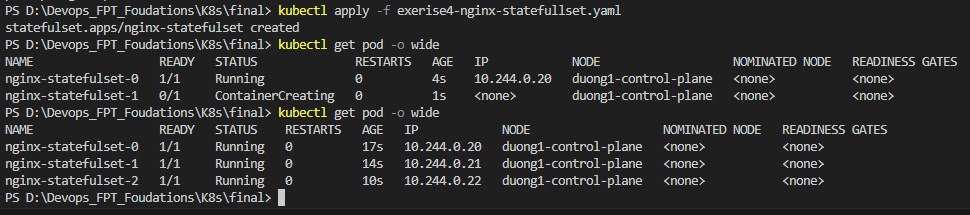


4 Create a nginx app with statefullset (Example)(1.5)

Chúng ta tạo pod nginx với statefullset với replica là 3 có service, volume với tên là nginx-data

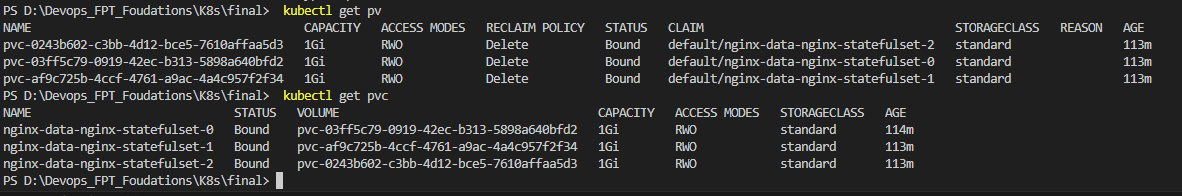


Ta apply và đợi xem kết quả



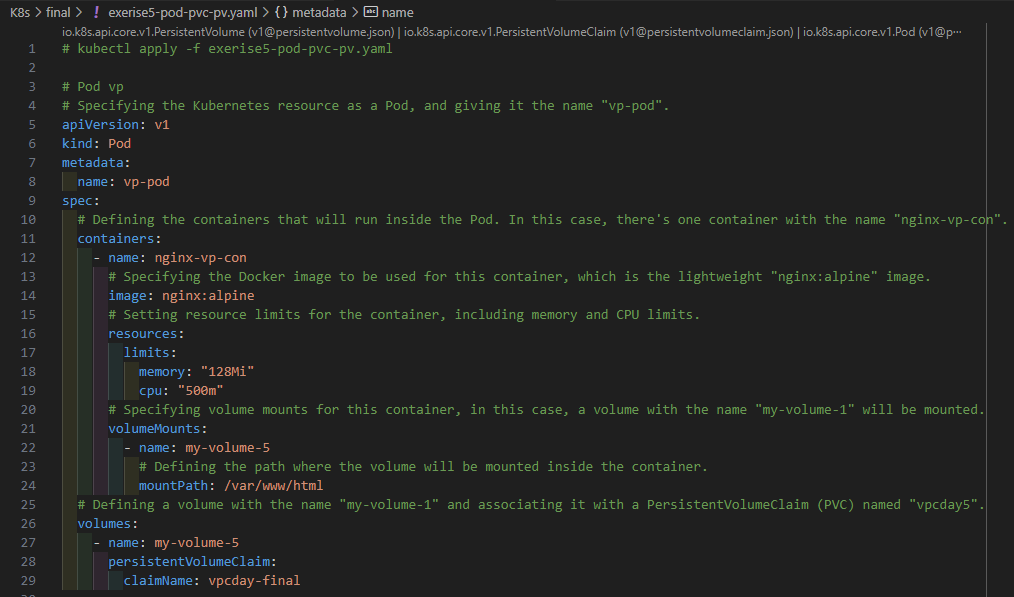
Ta đã tạo được 3 replica container nginx đều có port là 80 và được gán volume tên là nginx-data, volume này yêu cầu sức chứa là 1Gb

Ta cũng tạo được 3 pv và 3 pvc theo yêu cầu của yaml statefullset

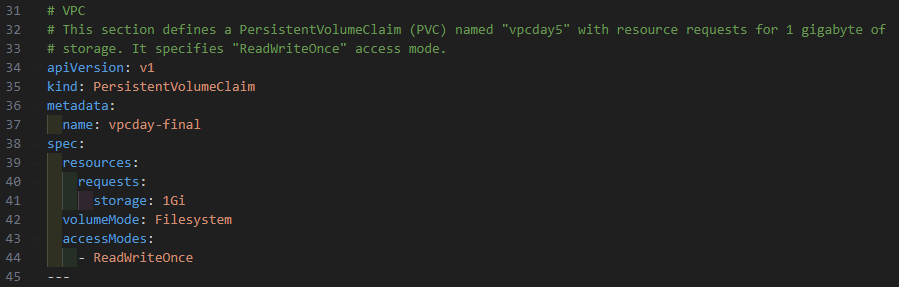


5 Create a nginx app using Persisten Volumes Claim to claim Persisten Volume resource (1.5)

Ta viết file yaml config pod se được gán vào một Persisten Volumes Claim tên my-volume-5



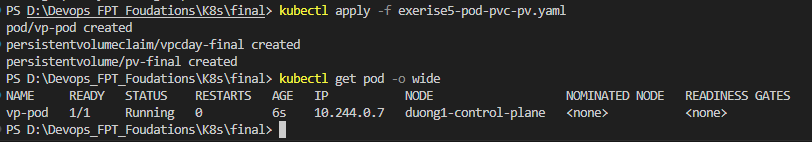
File config yaml của my-volume-5 với yêu cầu 1Gb và sau đó nó sẽ tìm 1 Persisten Volumes đáp ứng



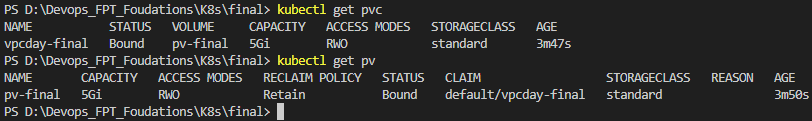
Ta tạo file Persisten Volumes được mount trên node host file có sức chứa là 5gb



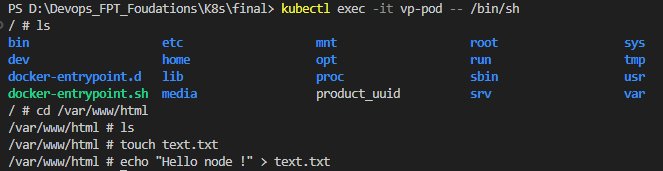
Ta apply



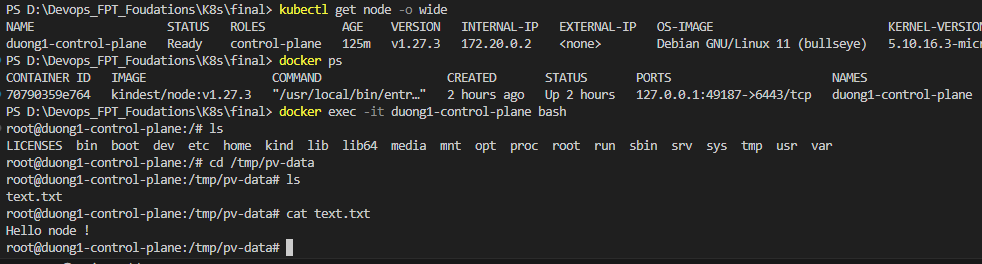
Ta sau đó check pv và pvc



Ta vào container trong pod và viết vào đường dẫn pvc volume (/var/www/html) file text.txt



File này theo đường dẫn sẽ được đưa vào Persisten Volumes Claim và truyền thẳng vào node theo đường dẫn “/tmp/pv-data” nhờ vào thằng Persisten Volumes đã được mount vào node file systems



Theo như kết quả trên ta vào liệt kê node và sau đó vào thẳng node chứa pod chúng ta mới tạo đang chạy ở đây là duong1-control-plane sau đó chúng ta vào thẳng đường dẫn tmp/pv-data để check file

Text.txt chúng ta đã tạo khi ở trong container pod. File truyền qua PVC PV rồi vào file của Node