1. Create a YAML manifest for a Pod named complex-pod. The main application container named app should use the image nginx and expose the container port 80. Modify the YAML manifest so that the Pod defines an init container named setup that uses the image busybox. The init container runs the command

“wget -o- google.com”

kubectl run complex-pod --restart=Never --port=80 --image=nginx:1.19.0 --dry-run=client -o yaml > complex-pod.yaml

Edit using vim and add the init container definition:

|  |
| --- |
| apiVersion: **v1**  kind: **Pod**  metadata:  creationTimestamp: **null**  labels:  run: **complex-pod**  name: **complex-pod**  spec:  containers:  - image: **nginx:1.19.0**  name: **app**  ports:  - containerPort: **80**  resources: {}  initContainers:  - name: **setup**  image: **busybox:latest**  command: **[ "sh", "-c", "wget -o- www.google.com"]**  dnsPolicy: **ClusterFirst**  restartPolicy: **Never**  status: {} |

1. Create the Pod from the YAML manifest

k apply –f complex-pod.yaml

1. Download the logs of the init container. You should see the output of the

wget command.

k logs complex-pod -c setup

1. Open an interactive shell to the main application container and run the

ls command. Exit out of the container.

k exec -it complex-pod -c app -- /bin/sh

1. Force-delete the Pod

k delete pods complex-pod --grace-period=0 –force

1. Create a YAML manifest for a Pod named data-exchange. The main application container named main-app should use the image busybox. The container runs a command that writes a new file every 30 seconds in an infinite loop in the directory /var/app/data. The filename follows the pattern {counter++}-data.txt. The variable counter is incremented every interval and starts with the value 1.

k run data-exchange --image=busybox --restart=Never --dry-run=client -o yaml > data-exchange.yaml

1. Modify the YAML manifest by adding a sidecar container named sidecar. The sidecar container uses the image busybox and runs a command that counts the number of files produced by the main-app container every 60 seconds in an infinite loop. The command writes the number of files to standard output.
2. Define a Volume of type emptyDir. Mount the path /var/app/data for both containers.

|  |
| --- |
| apiVersion: **v1**  kind: **Pod**  metadata:  creationTimestamp: **null**  labels:  run: **data-exchange**  name: **data-exchange**  spec:  containers:  - image: **busybox:latest**  name: **main-app**  command: [**'sh'**,**'-c'**,  **'index=1;**  **echo $index;**  **while true;**  **echo "Index : $index";**  **do touch /var/app/data/$index-data.txt;**  **index=$((index+1));**  **sleep 10;**  **done'**]  resources: {}  volumeMounts:  - mountPath: **/var/app/data**  name: **data-dir**  - image: **busybox:latest**  name: **sidecar**  command: [**'sh'**, **'-c'** ,  **'while true;**  **echo "checking files ...";**  **do ls -1 /var/app/data/ | wc -l;**  **sleep 5;**  **done'**]  volumeMounts:  - mountPath: **/var/app/data**  name: **data-dir**  volumes:  - name: **data-dir**  emptyDir: {}  dnsPolicy: **ClusterFirst**  restartPolicy: **Never**  status: {} |

1. Create the Pod. Tail the logs of the sidecar container.

k apply -f data-exchange.yaml

1. Delete the pod.

k delete pods data-exchange