**Lab-7: State Persistence – Multi-container Patterns**

(3 marks)

1. Create a Pod YAML file with two containers that use the image alpine Provide a command for both containers that keep them running forever.
2. Define a Volume of type emptyDir for the Pod. Container 1 should mount the Volume to path /etc/a, and container 2 should mount the Volume to path /etc/b.
3. Open an interactive shell for container 1 and create the directory data in the mount path. Navigate to the directory and create the file hello.txt with the contents “Hello World.” Exit out of the container.
4. Open an interactive shell for container 2 and navigate to the directory /etc/b/data. Inspect the contents of file hello.txt. Exit out of the container.

(4 marks)

1. Create a PersistentVolume named logs-pv that maps to the hostPath /var/logs. The access mode should be ReadWriteOnce. Provision a storage capacity of 5Gi. Ensure that the status of the PersistentVolume shows Available.
2. Create a PersistentVolumeClaim named logs-pvc. The access it uses is ReadWriteOnce. Request a capacity of 2Gi. Ensure that the status of the PersistentVolume shows Bound.
3. Mount the PersistentVolumeClaim in a Pod running the image nginx at the mount path /var/log/nginx.
4. Open an interactive shell to the container and create a new file named my-nginx.log in /var/log/nginx. Exit out of the Pod.
5. Delete the Pod and re-create it with the same YAML manifest. Open an interactive shell to the Pod, navigate to the directory /var/log/nginx, and find the file you created before.
6. Run the Init Container pattern example (1 mark)
7. Run the web-app.yaml (Sidecar pattern first example). What is the difference between this and the one in step 10? (1 mark)
8. Run the sidecar.yaml (Sidecar pattern second example). Use the commands in the example.txt file to demo how it works:

kubectl create -f sidecar.yaml

kubectl get pods webserver

kubectl logs webserver -c sidecar

kubectl exec webserver -it -c sidecar -- /bin/sh

wget -O- localhost?unknown

cat /var/log/nginx/error.log

(1 mark)