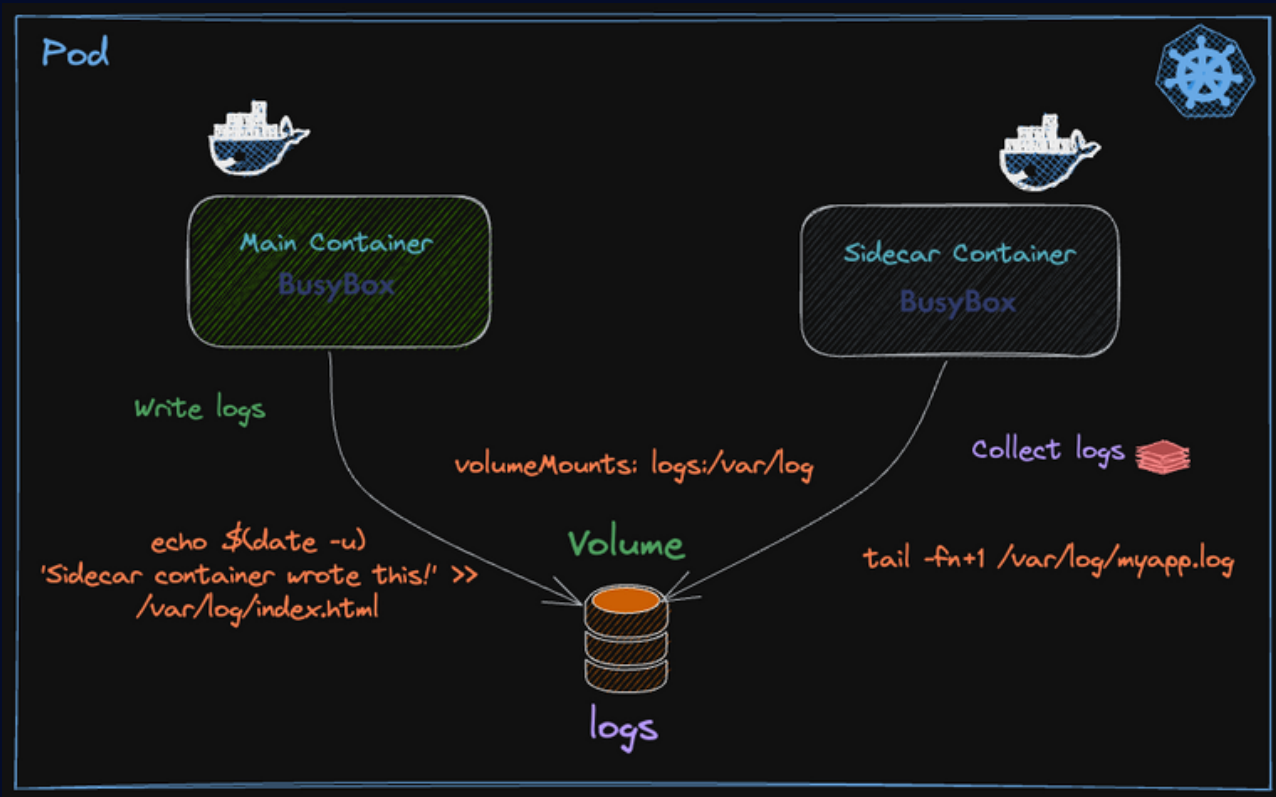




# Sidecar Container Pattern

Sidecar containers are containers that are required for the main container to function fully. The two containers share resources, such as pod storage.



In general, the sidecar supports functions required by the main application, for example, the main application can write registers and the sidecar can read them for debugging purposes.

Pod definition

Main Container

Sidecar Container

Shared Volume

```
apiVersion: v1
kind: Pod
metadata:
  name: test-sidecar
spec:
  containers:
    - image: busybox
      name: main-container
      args: ["sh", "-c", "while true; do echo $(date -u) 'Main container wrote this!' >> /var/log/myapp.log; sleep 5; done"]
      volumeMounts:
        - name: logs
          mountPath: /var/log
    - image: busybox
      name: sidecar-container
      args: ["sh", "-c", "tail -fn+1 /var/log/myapp.log"]
      volumeMounts:
        - name: logs
          mountPath: /var/log
  volumes:
    - name: logs
      emptyDir: {}
```

Prints the current date/time, with a message every 5 seconds in myapp.log file.

Display the last line of a text file and then monitor the file for updates

You can see the logs of the sidecar-container to read the file  
`kubectl logs test-sidecar -c sidecar-container`