Bash Script to Delete Log Files Inside Pods Based on Retention Days

```bash
#!/bin/bash

# Retention days for log files (delete files older than this many days)
RETENTION_DAYS=2  # Change this to your desired retention period in days

# Directory where the logs are mounted (should be the same as mountPath in Kubernetes)
LOG_DIR="/Log"  # Change this to the path where logs are mounted on your system

# Namespace where the pods are located
NAMESPACE="<namespace>"  # Change this to your actual namespace

# Patterns to search for in pod names
PATTERNS=("<pattern-1", "<pattern-2", "<pattern-n")

# Find all pod names that match the multiple patterns in the specified namespace
PODS=$(kubectl get pods -n "$NAMESPACE" --no-headers -o custom-columns=":metadata.name" | grep -E "$PATTERN_REGEX")

# Check if there are any matching pods
if [ -z "$PODS" ]; then
  echo "No pods found matching the patterns in namespace $NAMESPACE."
  exit 1
fi

# Loop through each pod name in the array
for POD in "${PODS[@]}"
do
  echo "Cleaning logs in pod $POD..."
```

```
  # Execute the command to delete log files in the pod older than retention days
  kubectl exec -n "$NAMESPACE" "$POD" -- /bin/sh -c "find $LOG_DIR -type f -name 'town_*.log' -mtime
+$RETENTION_DAYS -exec rm -f {} \;"
  kubectl exec -n "$NAMESPACE" "$POD" -- /bin/sh -c "find $LOG_DIR -type f -name 'town*.log' -mtime
+$RETENTION_DAYS -exec rm -f {} \;"


  # Confirmation message
  echo "Log cleanup complete for pod $POD."
 done
```