

GOOD LOGS

STRUCTURED LOGGING

STRUCTURED LOGS

A terminal window with a dark background. The title bar shows 'cedi@mae:~' and standard window controls. The terminal content shows a command to filter logs for errors with a 403 status, followed by a single log entry. The log entry is a JSON object with fields for level, timestamp, caller, table name, org ID, message, and error. The message describes an index set problem. The error message is 'open /var/loki/boltdb-shipper-cache/loki_index_19498/fake: no such file or directory'. The terminal also shows a second prompt line with a cursor.

```
cedi@mae ~  
$ kubectl logs -n observability loki-backend-0 | grep "error" | grep -v "403" | tail -n 1  
level=error ts=2023-05-22T14:01:24.044644417Z caller=table.go:342 table-name=loki_index_19498 org_id=fake msg="index set fake has some problem, cleaning it up" err="open /  
var/loki/boltdb-shipper-cache/loki_index_19498/fake: no such file or directory"  
  
cedi@mae ~
```

- ▶ Easy to read by humans & log aggregation systems
- ▶ Provides contextual information using labels
- ▶ Easy to filter by only selecting relevant labels