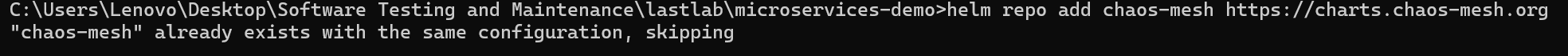
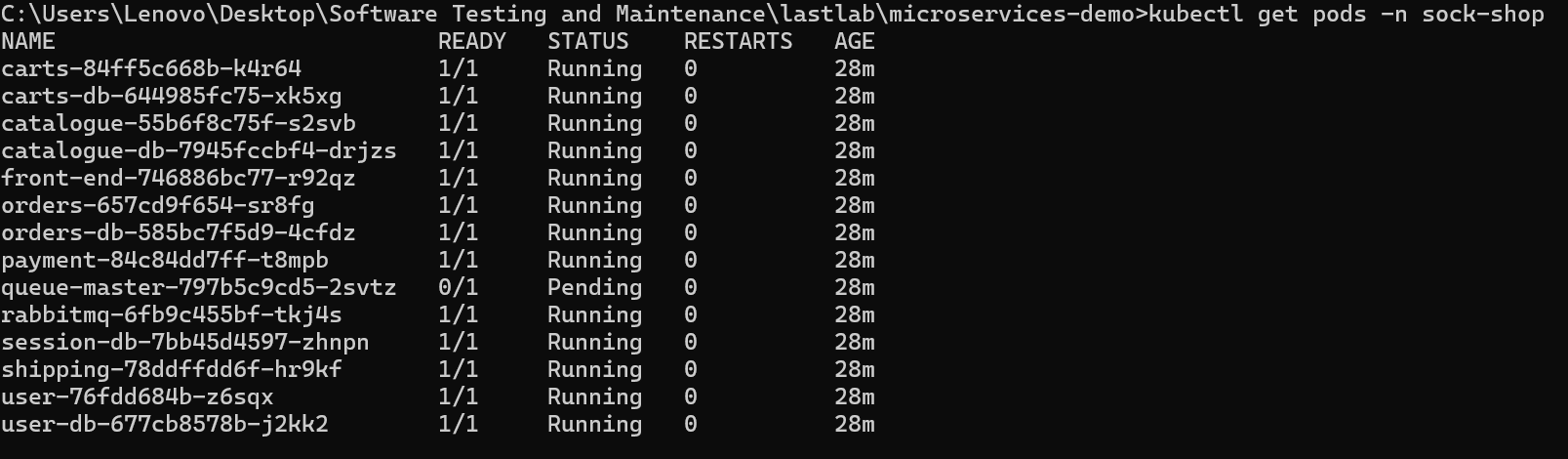
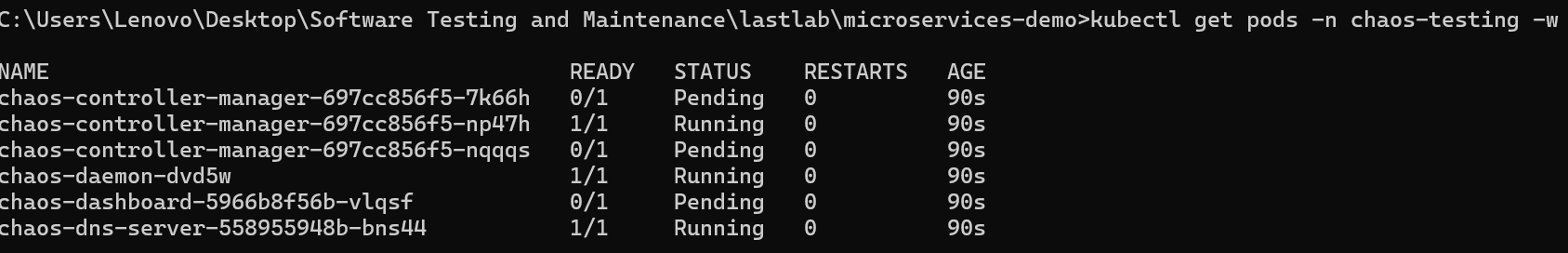
1. 安装和部署chaosmesh



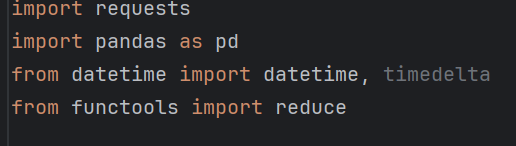
  
二、查看sock-shop微服务部署成功



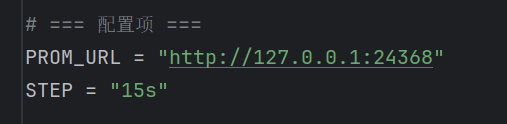
三、查看chaosmesh服务部署情况



四、编写python脚本，可以将对应数据导出

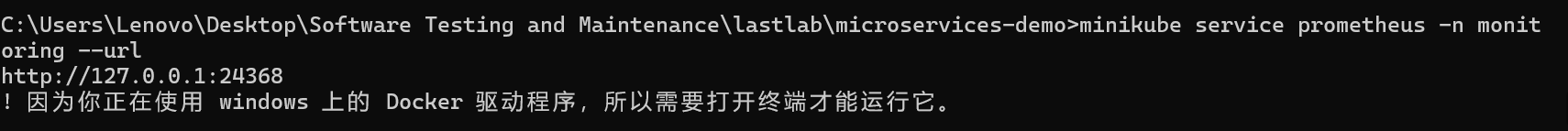


用于发起 HTTP 请求（requests），处理表格数据（pandas），时间处理（datetime），合并多个 DataFrame（reduce）。

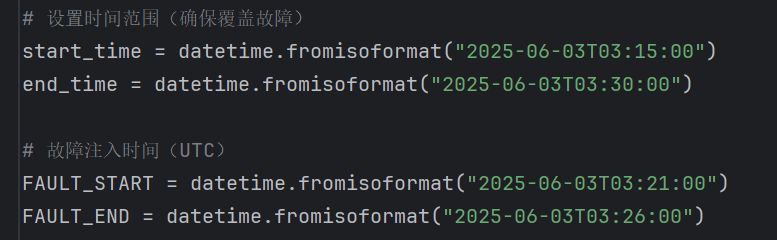


· Prometheus 本地服务地址

· STEP 表示查询粒度为每 15 秒一次



将普罗米修斯url填入



设置数据查询时间窗口，覆盖故障注入区间（故障持续时间）

Start\_time 查询开始时间

end\_time查询结束时间

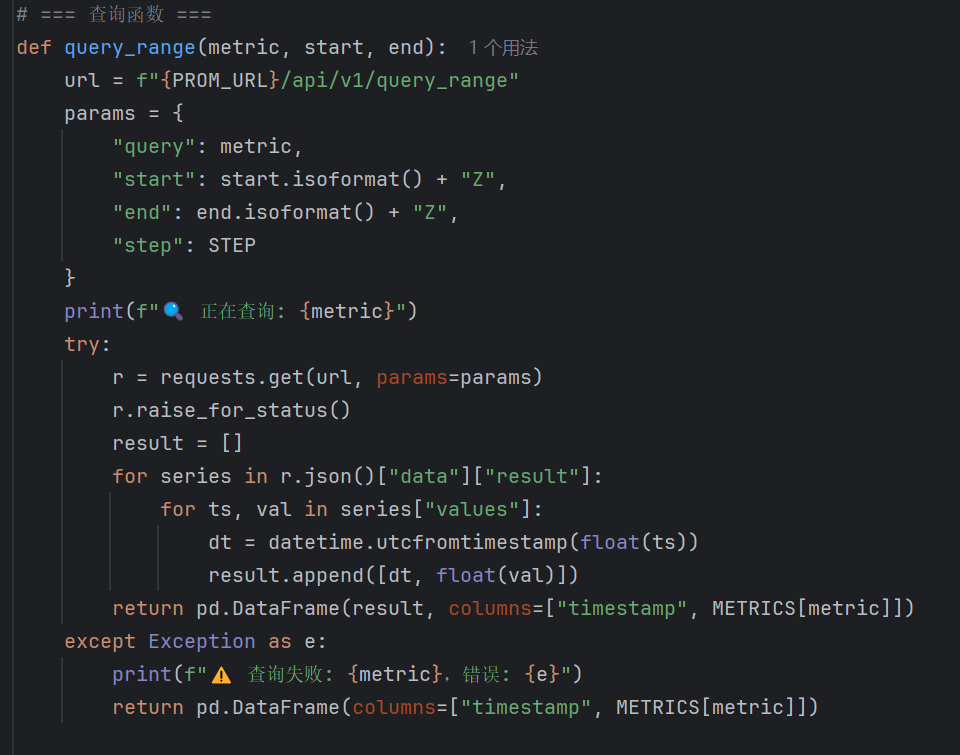
FAULT\_START故障注入开始时间

FAULT\_END故障注入结束时间



· 共有 25 个监控指标，涵盖 CPU、内存、线程、网络、磁盘、状态等

· 字典的 key 是 PromQL 查询语句，value 是输出 CSV 中的列名别名



该函数对每个指标发起 Prometheus 查询：

* 构造查询 URL 和参数
* 遍历响应结果中的时间序列数据（values 为 [timestamp, value]）
* 将每个指标结果转为 DataFrame，并返回

若失败则返回空 DataFrame 并打印错误信息。



批量执行每个指标的查询函数 query\_range

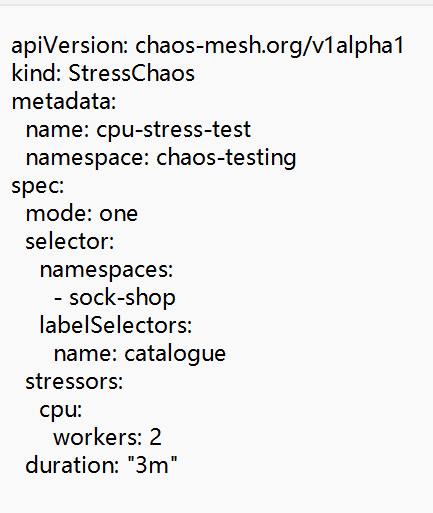
筛掉空的 DataFrame，避免 merge 出错

使用 reduce 将多个 DataFrame 依 timestamp 外连接合并成一个总表 df\_all

按时间排序、前向填补缺失值（fillna(method="ffill")），再去除仍有缺失的行（dropna）

1. 故障注入

（1）Yaml文件编写，故障类型为节点压力，节点选择catalogue，持续时间3分钟

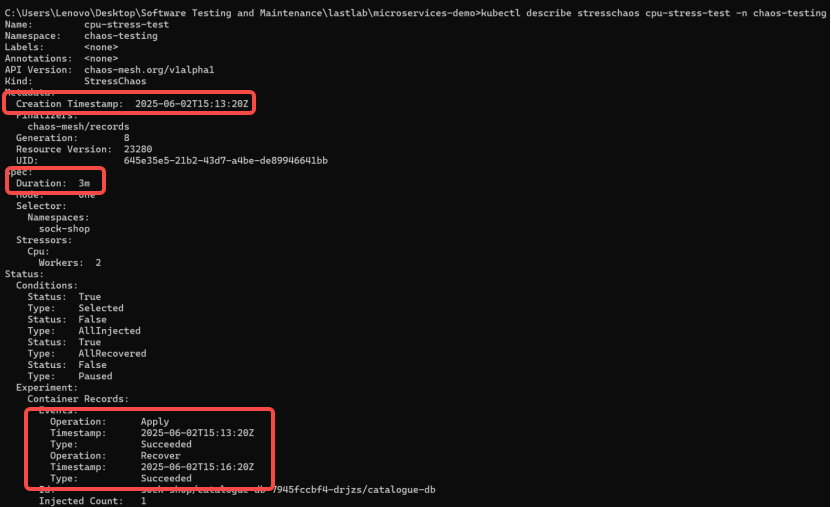
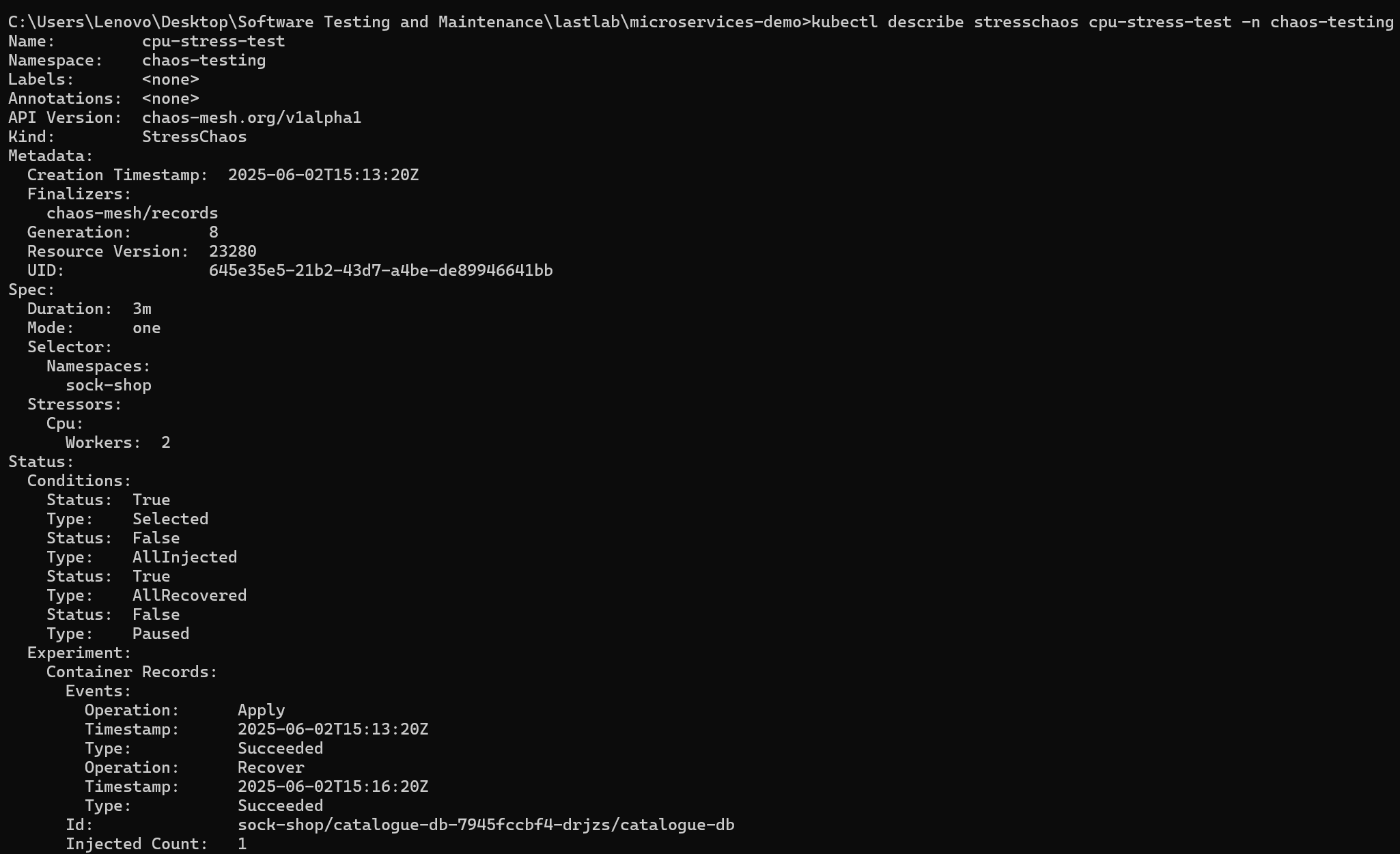


（2）

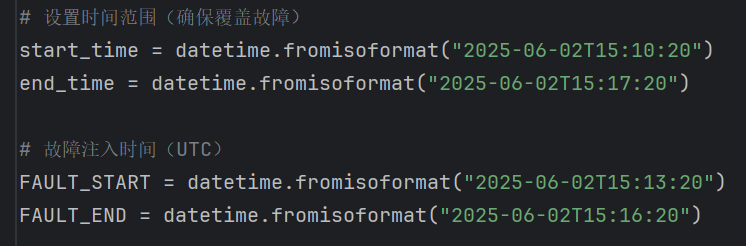
注入 catalogue Pod压力

apply 注入  

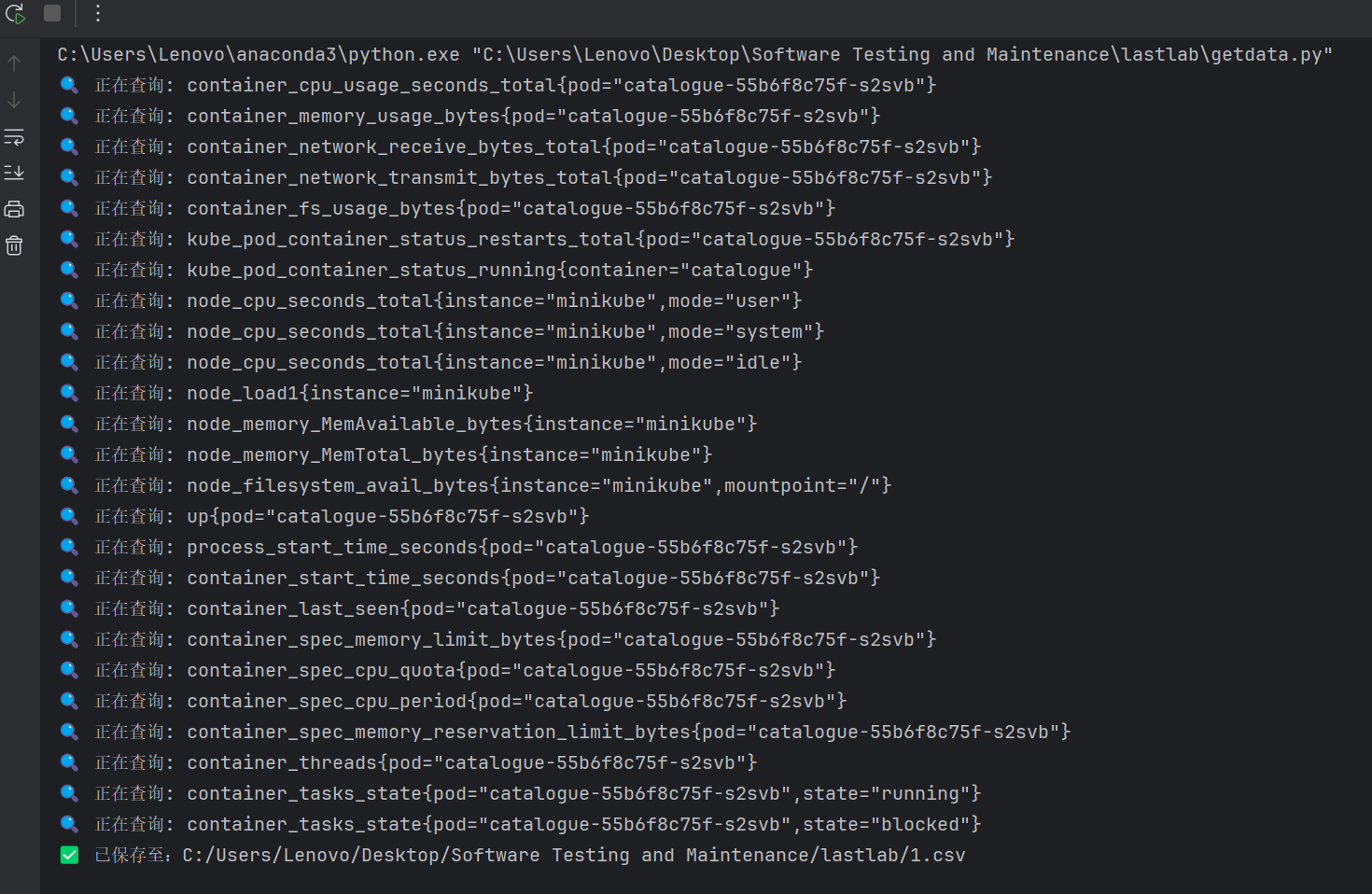

（3）查看注入情况，可以看到创建和指令操作时间，根据这个时间去python脚本进行设置



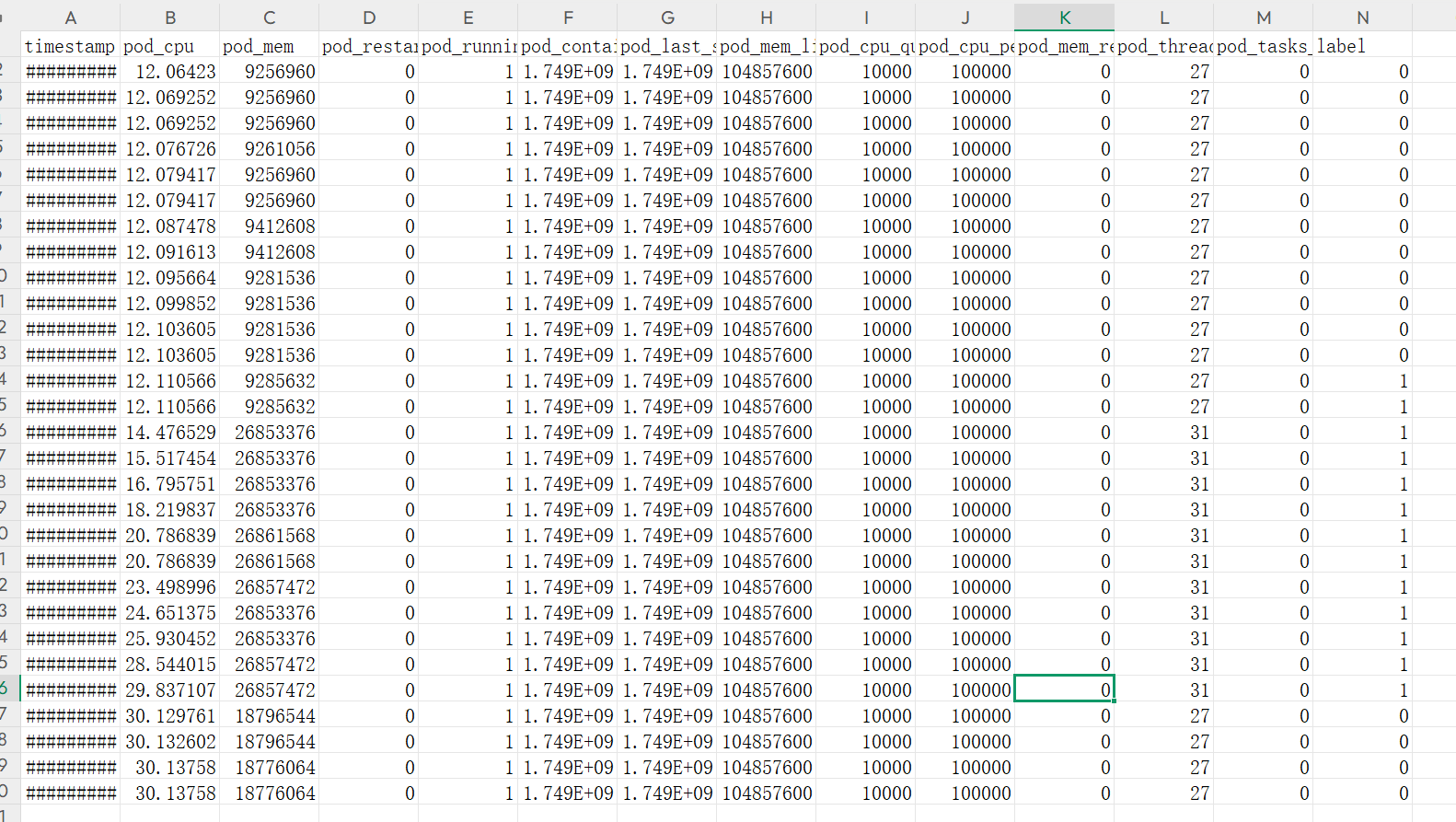
（4）开始时间和结束时间设置在前后7分钟，故障注入时间根据刚刚查到信息填入，因故障持续三分钟，所以结束时间顺延3分钟



（5）运行脚本

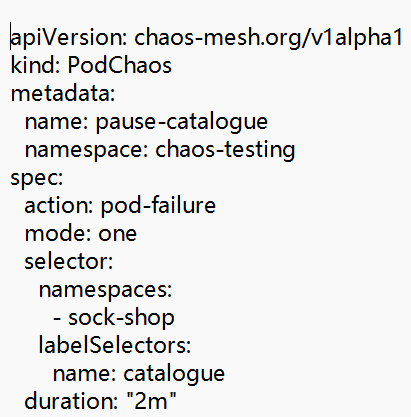


（6）数据如下显示

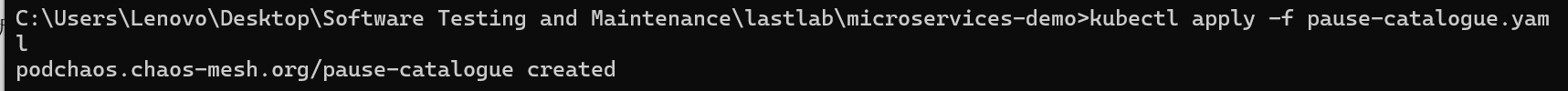


（1）

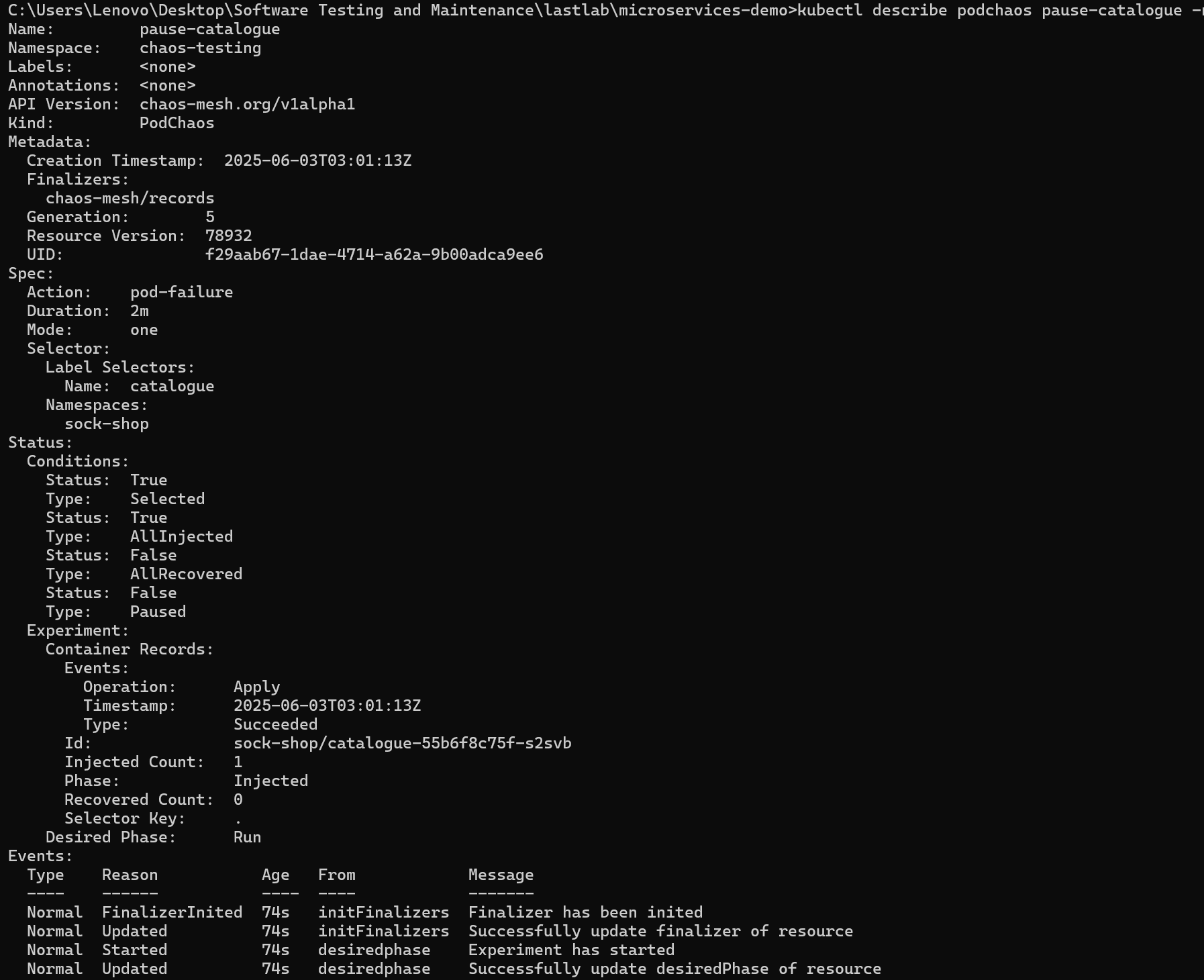
这次故障为暂停pod运行pod-failure（模拟网络级别不可达），暂停的pod为catalogue，持续时间2分钟



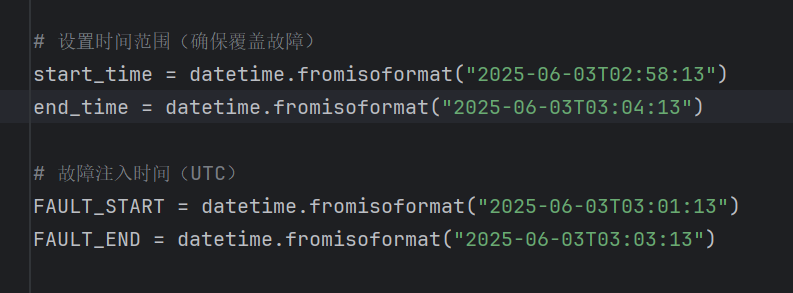
（2）注入



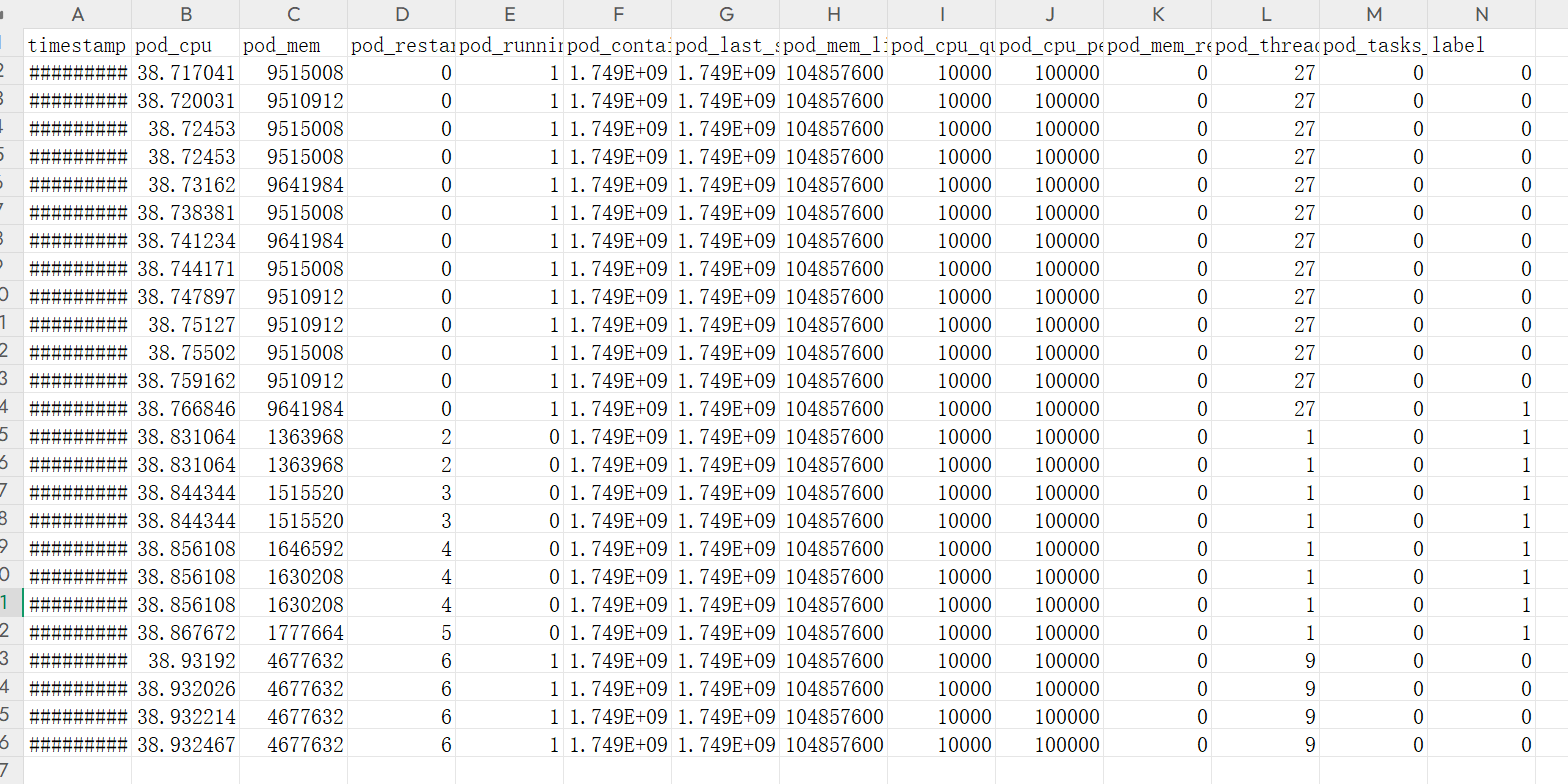
（3）查看注入情况，获取注入开始时间



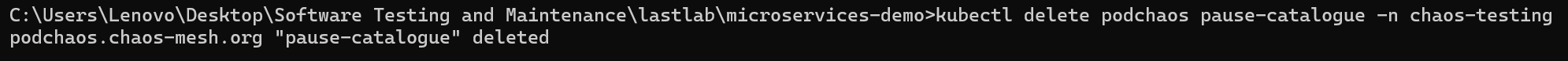
（4）修改脚本对应信息



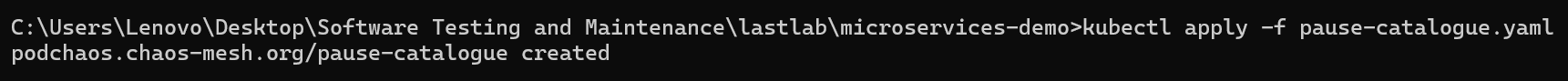
（5）导出数据如下

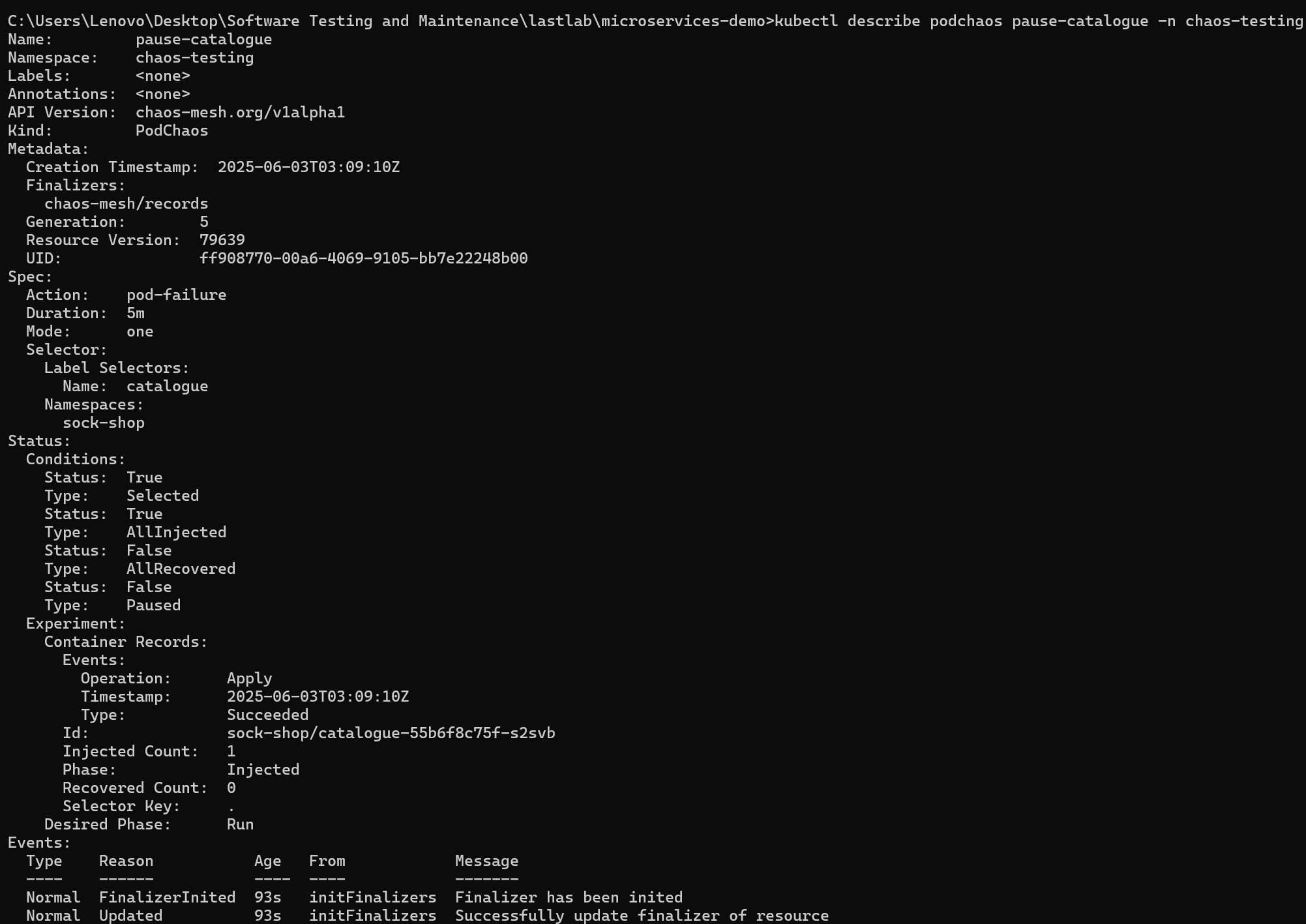


（6）删除已注入的 PodChaos

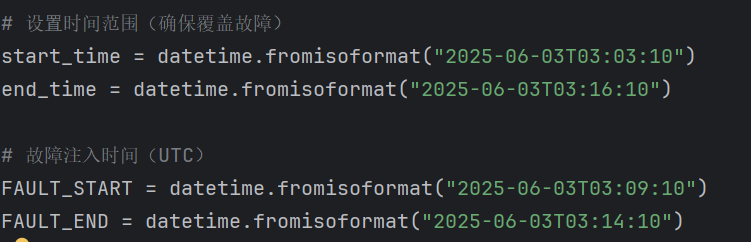


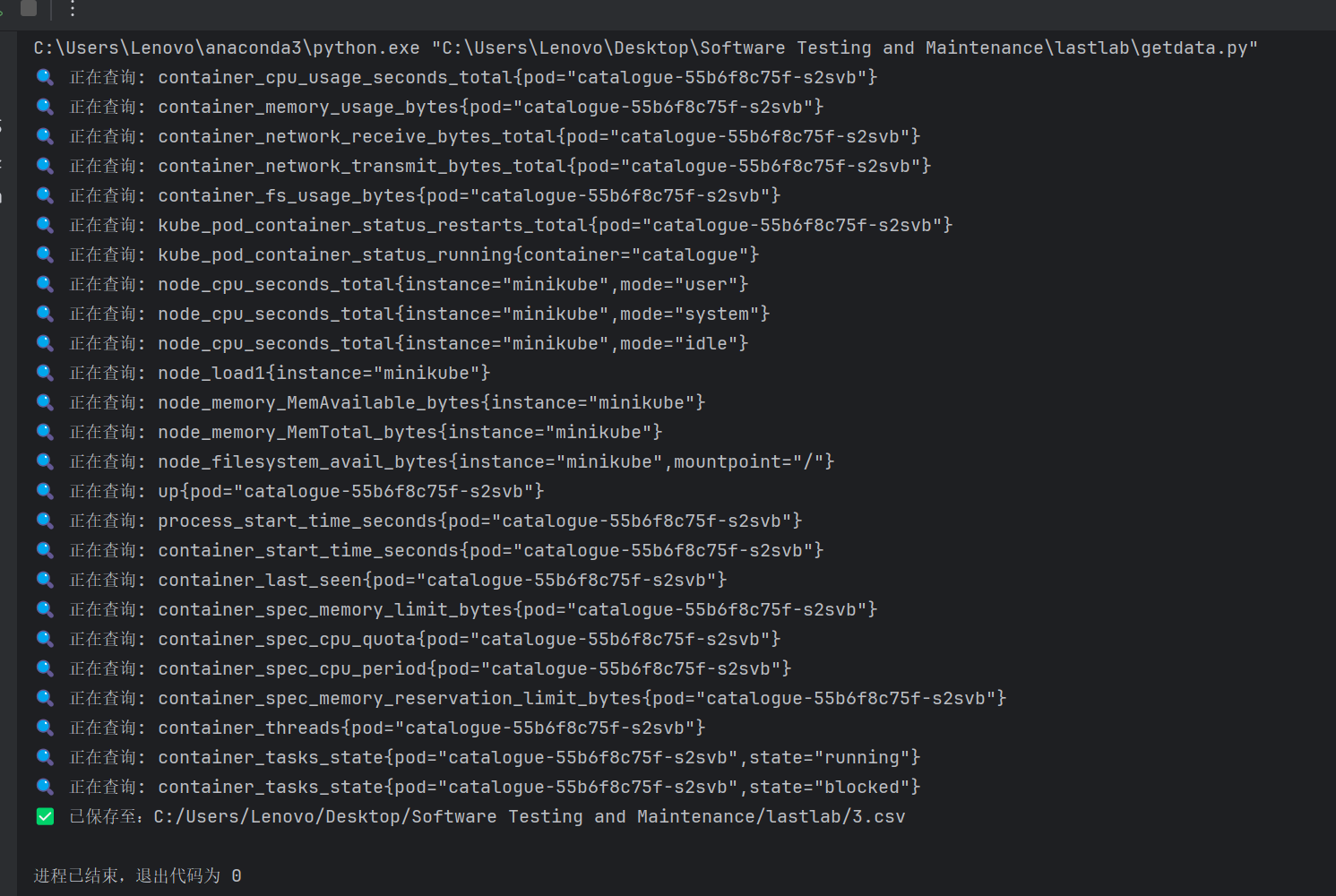
（7）重新注入pod暂停故障



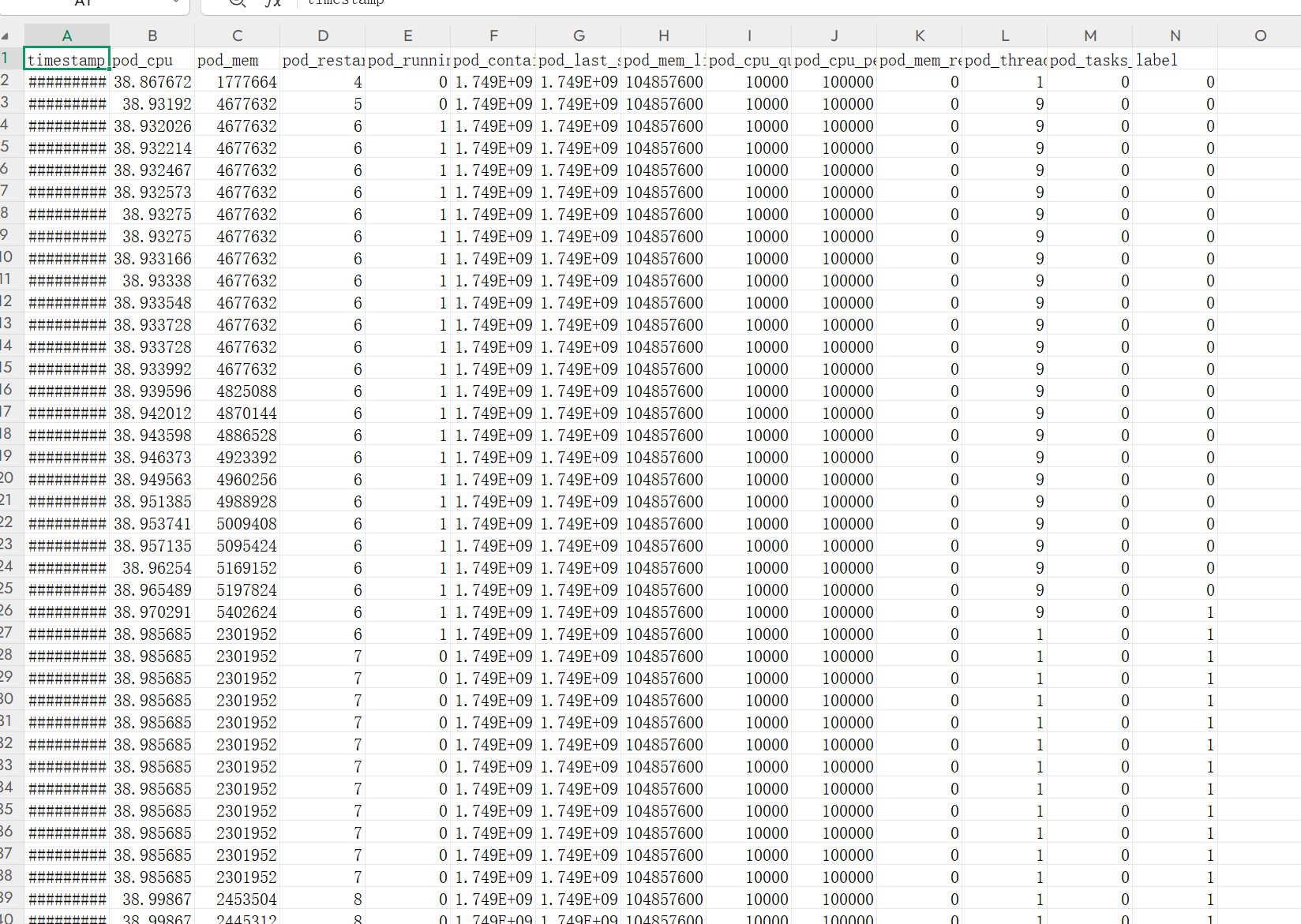
（8）查看状态

（9）设置时间

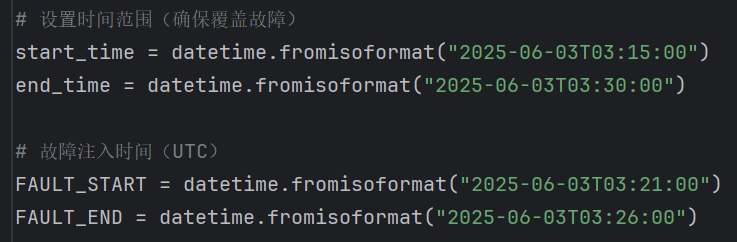
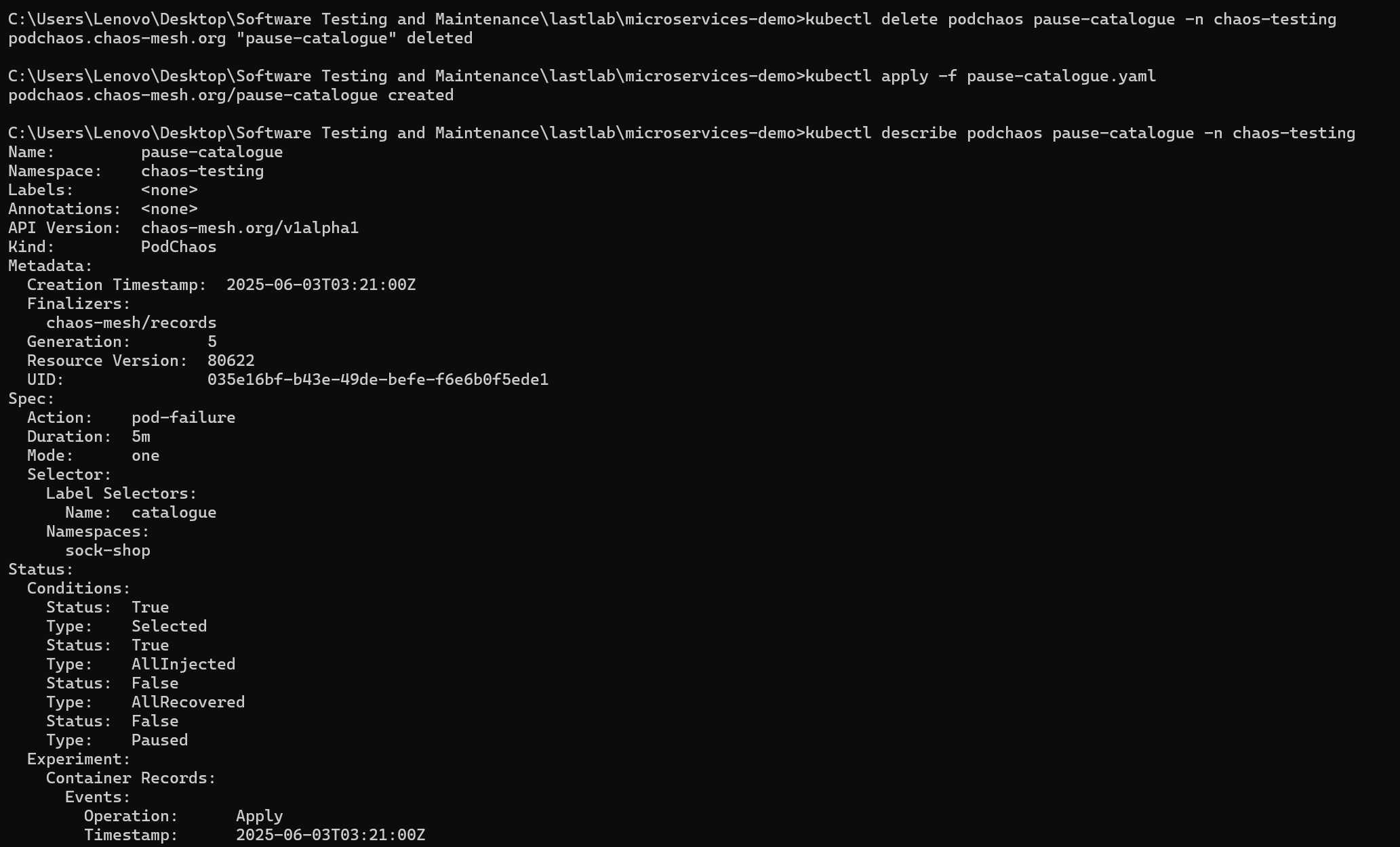


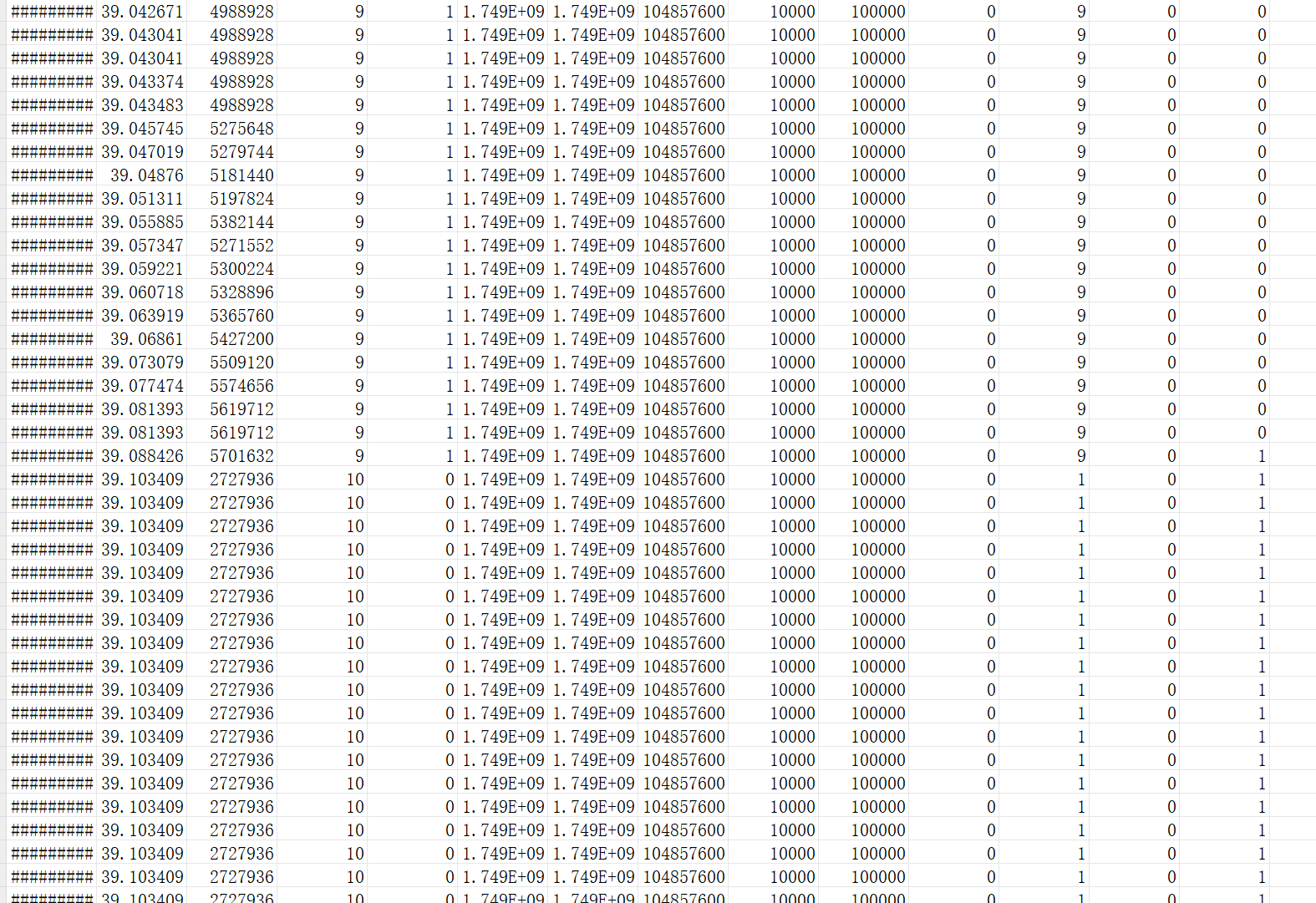


（10）导出数据

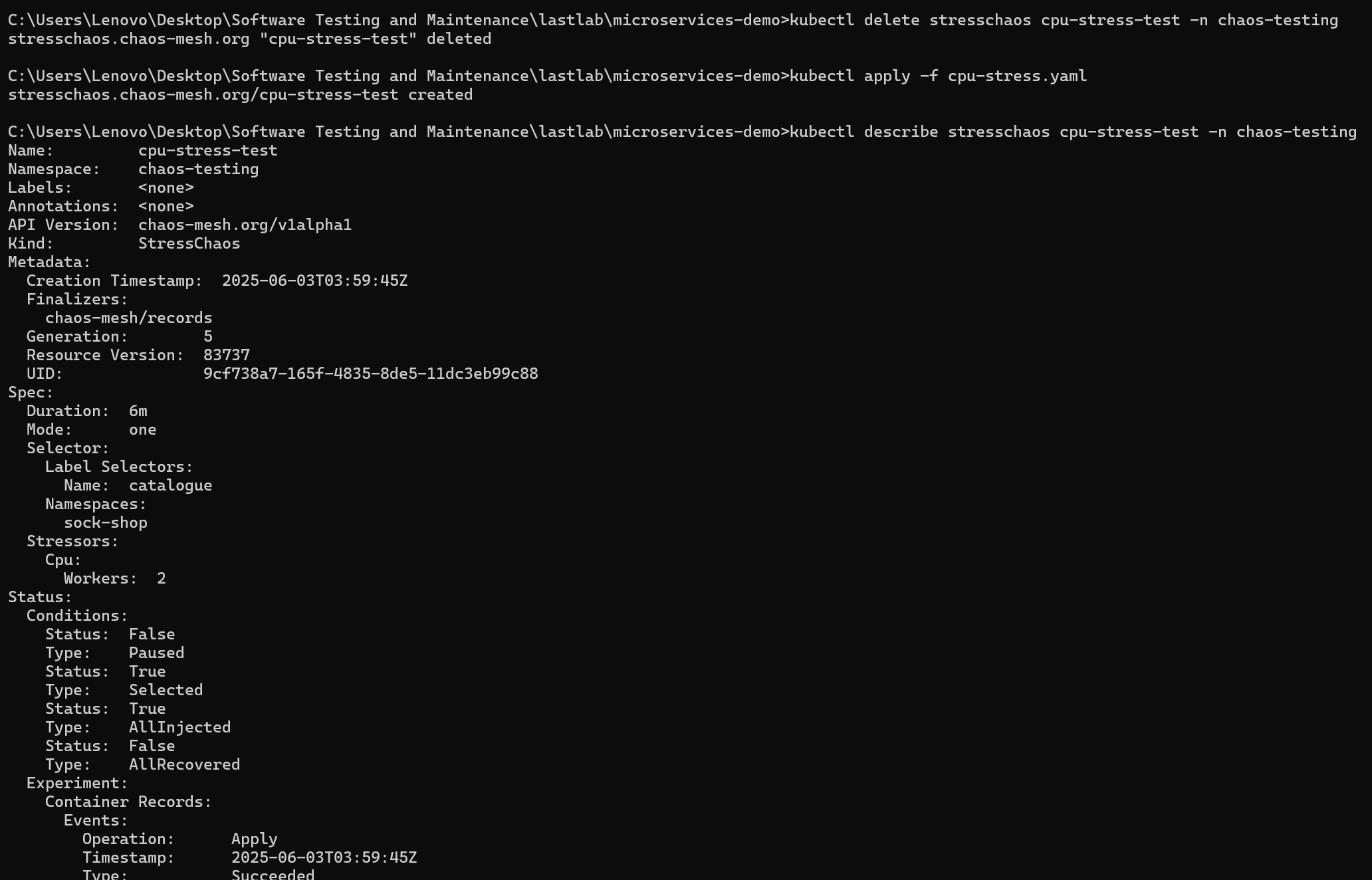


（11）继续循环此操作





（12）继续进行节点压力型故障的注入



#注：下面是注入payment的节点压力，但是数据不明显，就没有导出数据

