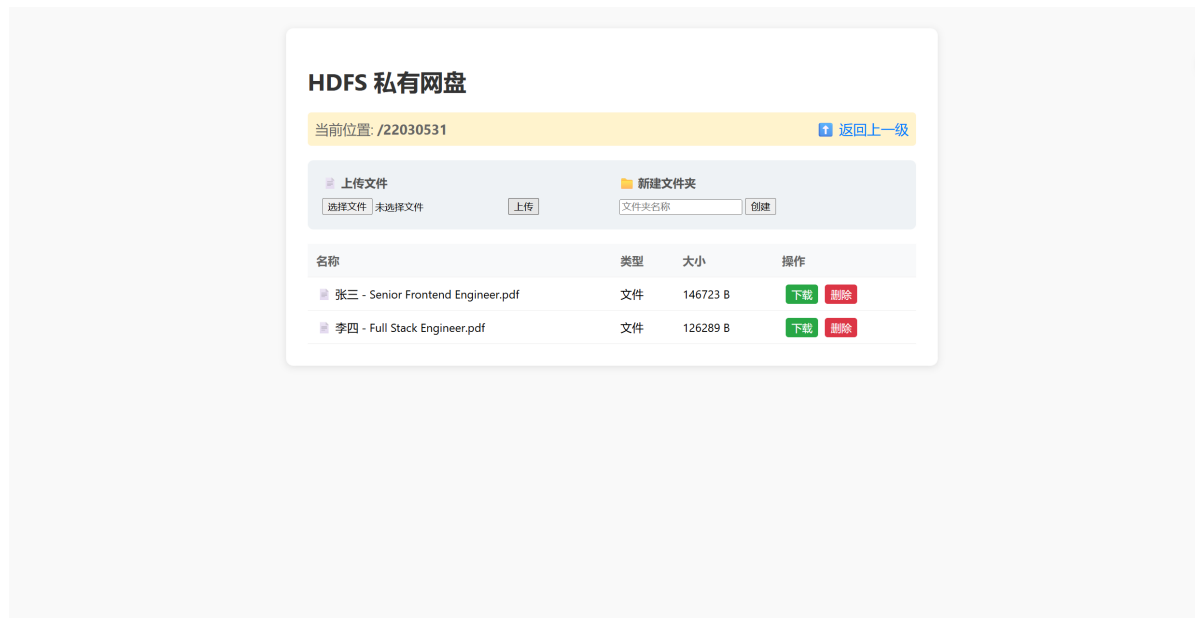


hdfs

<http://34.92.75.134:5000>

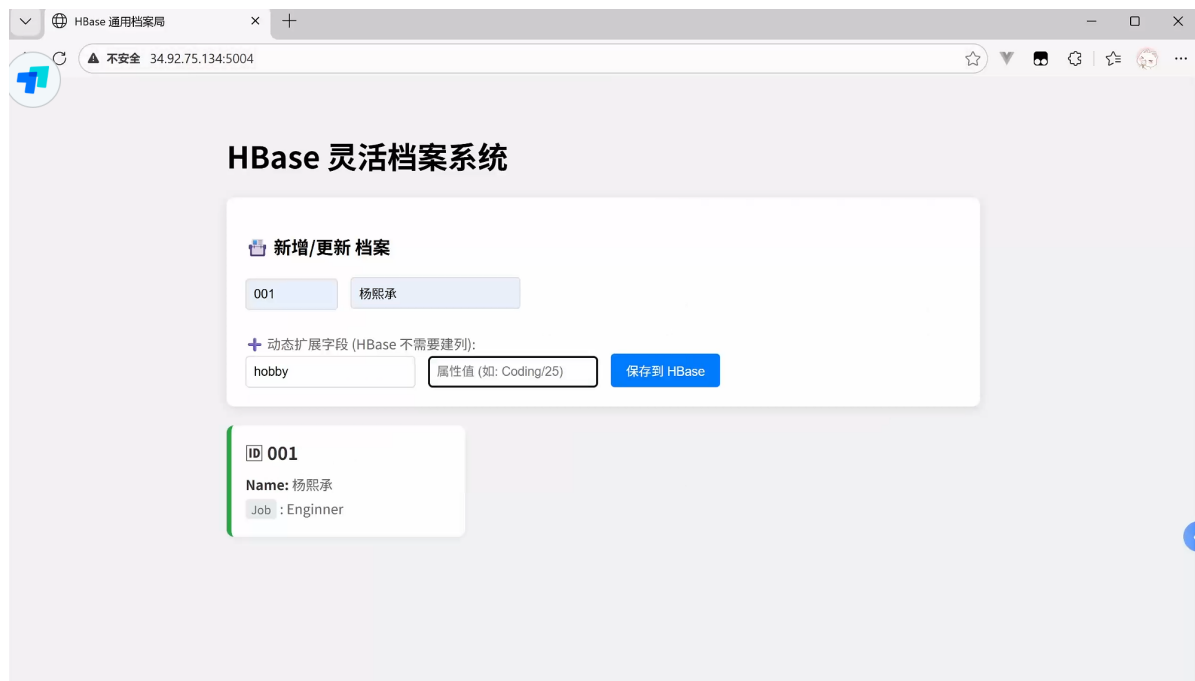
用 `docker-compose` 启动 **HDFS 伪集群**（1 个 NameNode + 多个 DataNode）、RPC、Flask Web 全部保持不变。Web 后端继续用 `hdfs.InsecureClient('http://namenode:9870')` 走 WebHDFS 做文件操作。页面支持目录浏览、上传、建目录、下载、删除（`/upload` `/mkdir` `/download` `/delete`）。

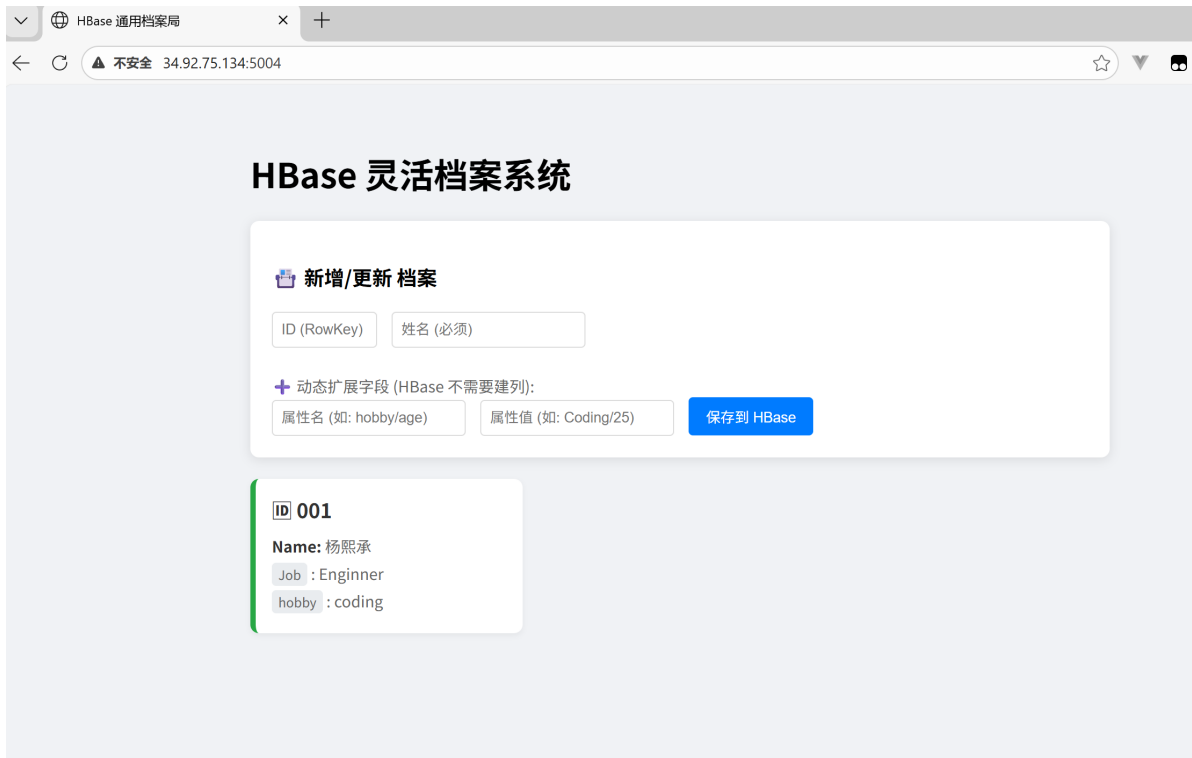


hbase

<http://34.92.75.134:5004>

用 `docker-compose` 启动 **HBase 伪集群**（ZooKeeper 多节点 + HMaster + 多 RegionServer + 独立 Thrift 服务）；同时 Flask Web 映射 5004。Web 后端用 `happybase` 连 Thrift（9090），启动时自动建 `profiles` 表并进行读写。页面功能不变：scan 展示用户列表 + 表单新增用户（含动态字段 `extra_key/extra_val`）。

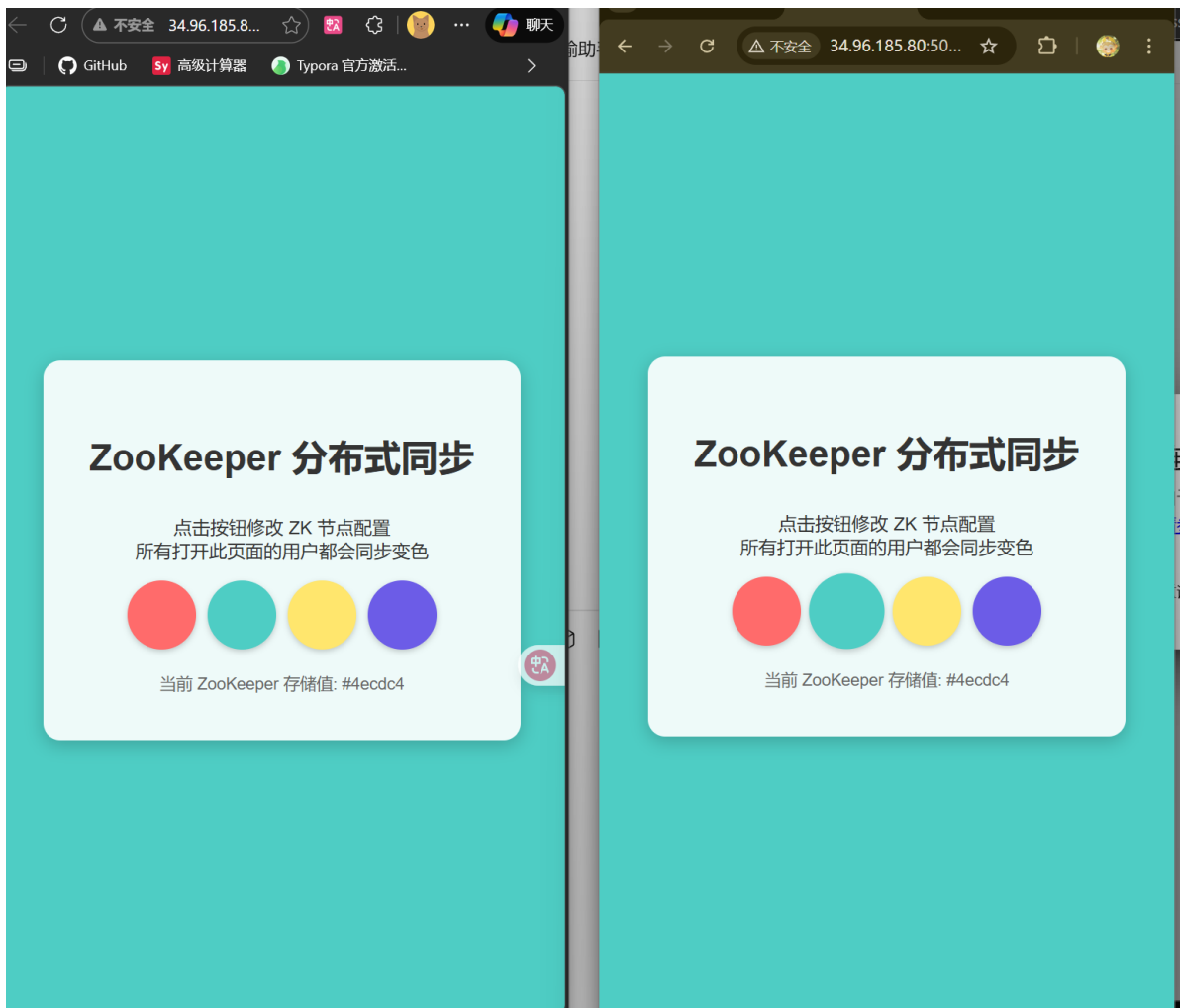




zookeeper

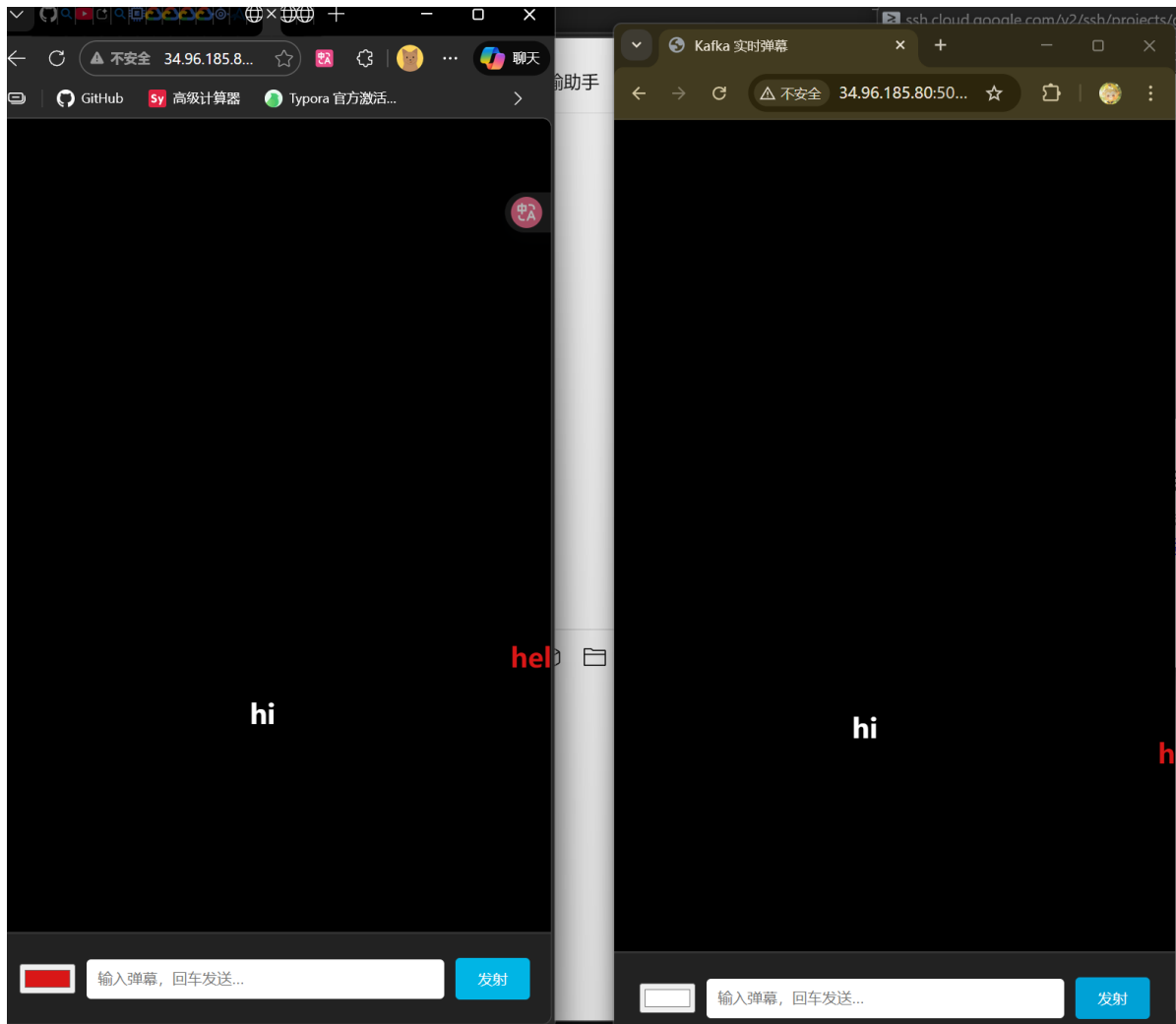
<http://34.92.75.134:5001>

用 `docker-compose` 启动 **ZooKeeper 伪集群 (3 节点)** + 一个 Flask Web，对外仍保留 `2181` 不变。Web 后端用 `kazoo` 连接 **ZK quorum**（例如 `zk1:2181,zk2:2181,zk3:2181`），把颜色值存到 `/config/color`。页面提供读取/设置颜色，通过 `/get_color` 和 `/set_color` 实现配置中心效果。



<http://34.92.75.134:5002>

用 `docker-compose` 启动 **ZooKeeper 伪集群 (3 节点)** + **Kafka 伪集群** + Flask Web, FlaskWeb 后端, Producer 发消息到 `danmaku_topic`, Consumer 实时消费并用 SSE 推送。页面功能不变: 发送弹幕 (文字+颜色) 和实时弹幕流 (`/send` + `/stream`)。



Flink

<http://34.92.75.134:5003>

用 `docker-compose` 启动 **Flink 伪集群** (1 个 JobManager + 多个 TaskManager) ; 同时构建 Flask Web (容器内 5000 映射宿主 5003)。Web 通过 `nc -lk 9999` 提供输入流端口, 用户点击购买 POST 到 `/buy` 写入数据供 Flink 作业消费。页面通过 `/stream` SSE 实时读取日志输出并展示“实时大屏”。

