Documentation as Exposee

Introduction

For this project, I selected Apache HBase as the database to achieve granular data access. HBase is a distributed, scalable NoSQL database built on Hadoop, designed to manage large datasets. It supports access control at the table, column family, and column qualifier levels natively via the AccessController coprocessor. Additionally, HBase offers cell-level access control through the VisibilityController coprocessor, allowing permissions to be set on individual cells (specific row-column intersections) using visibility labels (e.g., “public” or “secret”). This project implements a basic access control system with HBase, achieving qualifier-level granularity (e.g., restricting access to info:name), though not full cell-level control due to setup constraints.

> [!NOTE] True cell-level access control (e.g., restricting to specific rows like row1) requires the VisibilityController coprocessor, which attaches visibility labels to cells and enforces access based on user authorizations. This is not implemented here due to limitations in the dajobe/hbase:latest image, which lacks this coprocessor. Without it, the AccessController used in this setup supports permissions only at the table, column family, or column qualifier level—not row or cell level. For more details, see the HBase Visibility Labels documentation.
>
> Additionally, the simple authentication method used here prevents testing as the integrator user via the shell or REST API (curl). A more robust system like Kerberos would enable full user separation and testing but is too complex for this project, requiring a Kerberos Key Distribution Center (KDC) and keytab configuration.

Exposee Documentation

Setup HBase with Docker

Follow these steps to set up HBase with Docker:

1. Pull image by running following in your terminal:

```
docker pull dajobe/hbase:latest
```

2a. Create a docker compose file docker-compose.yml:

```
services:
  zookeeper:
    image: zookeeper:3.8
    container_name: zookeeper
    ports:
      - "2181:2181"

  hbase:
    image: dajobe/hbase:latest
    container_name: hbase-distributed
    ports:
      - "16000:16000"  # HMaster
      - "16010:16010"  # HMaster UI
      - "16020:16020"  # RegionServer
    environment:
      - HBASE_CONF_hbase_security_authentication=simple
      - HBASE_CONF_hbase_security_authorization=true
      -
HBASE_CONF_hbase_coprocessor_region_classes=org.apache.hadoop.hbase.securi
ty.access.AccessController
      - HBASE_CONF_hbase_zookeeper_quorum=zookeeper
    volumes:
      - hbase-data:/hbase-data
      - ./hbase-site.xml:/opt/hbase/conf/hbase-site.xml
    depends_on:
      - zookeeper

volumes:
  hbase-data:
```

2b. Configure HBase for Authorization: Create a file named hbase-site.xml to enable security features:

```
<?xml version="1.0"?>
<configuration>
  <property>
    <name>hbase.security.authentication</name>
    <value>simple</value>
  </property>
  <property>
    <name>hbase.security.authorization</name>
    <value>true</value>
  </property>
  <property>
    <name>hbase.coprocessor.region.classes</name>

<value>org.apache.hadoop.hbase.security.access.AccessController</value>
  </property>
</configuration>
```

3. Start the Containers:

```
docker-compose up -d
```

4. Access HBase shell:

```
docker exec -it hbase-distributed hbase shell
```

Create tabels and insert Sample data:

Run below commands in the HBase shell to insert sample data:

```
create 'no_access_table', 'info'
put 'no_access_table', 'row1', 'info:name', 'Frede'
create 'read_only_table', 'info'
put 'read_only_table', 'row1', 'info:name', 'Frederik'
put 'read_only_table', 'row1', 'info:salary', '50000'
put 'read_only_table', 'row2', 'info:name', 'Daniel'
put 'read_only_table', 'row2', 'info:salary', '60000'
create 'read_write_table', 'info'
put 'read_write_table', 'row1', 'info:name', 'Anders'
```

Create user and grant permissions

This will create an user called Integrator with password integrator_pass and grant permissions to the tables. Below R only allows read access to the 'info:name' column in the 'read_only_table' table. RW allows read and write access to the 'read_write_table' table.

```
grant 'integrator', 'R', 'read_only_table', 'info', 'name'
grant 'integrator', 'RW', 'read_write_table'
```

Verify Permissions

To check access use `user_permission` followed by the table to check the permissions for the user. It should display the permissions for the user integrator to

```
user_permission 'read_only_table'
```

This will output:

```
User
Namespace,Table,Family,Qualifier:Permission
 integrator
default,read_only_table,info,name: [Permission: actions=READ]
```

Ideally then you would be able to access the integrator user but this is not possible with the HBase shell. Then you would run scan 'no_access_table' to validate integrator user does not have access to the table. Also run scan 'read_only_table' to validate integrator user only has access to the 'info:name' column.