CPSC 449 - Web Back-End Engineering

Exercise 2 - Fall 2023

This exercise may be completed individually, or in a pair with another student. If you work with another student, only one of you needs to submit via Canvas.

Exercise

In this exercise, you will experiment with two software packages needed for Project 2, the KrakenD API Gateway and the LiteFS distributed file system. KrakenD will act as a reverse proxy and provide JWT authentication, while LiteFS replicates a SQLite database.

Installation

Complete the following steps to install KrakenD and LiteFS. The instructions provided are for Tuffix 2022. In order to complete this exercise you will need the `fastapi/api` example from the `cpsc449` repository.

Installing KrakenD as an API gateway

Use the instructions for installing KrakenD from the Ubuntu or Debian DEB packages from the KrakenD download page:

```
sudo apt install ca-certificates gnupg
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv 5DE6FD698AD6FDD2
echo "deb https://repo.krakend.io/apt stable main" | sudo tee
/etc/apt/sources.list.d/krakend.list
sudo apt update
sudo apt install --yes krakend
```

Krakend will start with a global configuration located in `/etc/krakend`, but we want to use a local configuration file and start and stop krakend with Foreman. Stop and disable the global service.

```
sudo systemctl disable krakend --now
```

Installing entr to restart KrakenD

Unlike Uvicorn, KrakenD does not include a --reload switch to automatically restart the process when its configuration changes. If you would like to do this, install the Event Notify Test Runner.

```
sudo apt install --yes entr
```

Installing LiteFS to replicate the database

Download the latest AMD64 release of LiteFS, then extract the `litefs` binary and place it in `./api/bin`. Try running it, and verify that it prints a help message.

Configuring KrakenD to proxy API calls

Follow the instructions from Week 4 to start the Books API, and verify that you can access http://localhost:5000/books/.

Work through Running KrakenD server. Verify that you can run KrakenD with the `krakend run` command and that you can access http://localhost:8080/__health, Place the `krakend.json` file in `./etc`.

Define a KrakenD endpoint to `GET /api/books/`, with the backend http://localhost:5000/books/. Restart KrakenD and verify that you are able to access this endpoint at http://localhost:8080/api/books.

To restart KrakenD automatically when `./etc/krakend.json` changes, use the following command;

```
echo ./etc/krakend.json | entr -nrz krakend run --config etc/krakend.json
```

Define the remaining endpoints for the Books API, all mounted under the prefix `/api`. Note the following:

- KrakenD defaults to hiding HTTP status codes and error messages, so when a backend returns an HTTP status code that indicates an error, you will always see HTTP 500 Internal Server Error, with no further information.

  In order to see HTTP 409 Conflict and the error message from the database for the `POST /api/books` endpoint, see Strategies to return headers and errors, in particular Return backend errors in a new key.

- You will need to define a list of allowable query string parameters for the `GET /api/search` endpoint.

- When debugging KrakenD errors, it helps to see the Uvicorn access_log data. Check your copy of the cpsc449 repository to make sure that it has the latest change.

Adding KrakenD to the Procfile

Add a new process type to the Procfile to start KrakenD. Be sure to add the `--port $PORT` switch to allow Foreman to choose nonconflicting port numbers.

Configuring database replication with LiteFS

Create two LiteFS configuration files in `./etc` named `primary.yml` and `secondary.yml` and two sets of directories in `./var` with the following structure:

```
var
├── primary
│   ├── fuse
│   └── data
└── secondary
    ├── fuse
    └── data
```

Using the LiteFS Config Reference as a guide for both the primary and secondary replica configuration files, configure the following:

- The `fuse.dir` and `data.dir` for each replica
- For the primary replica, `exec` the `uvicorn` command from the `Procfile`.

  For the secondary replica, do not `exec` another process.
- The secondary replica's HTTP API Server should listen on a different port (e.g. `":20203"`).
- Configure static leasing.

Note: Once you have configured the `primary.yml` to `exec` the `uvicorn` command, do not attempt to run `./bin/litefs` directly unless you have first set the `PORT` environment variable.

Adding LiteFS to the Procfile

Remove the `api` process type from the `Procfile`, and create two new process types: one for the primary replica and one for the secondary. Each process type should run `bin/litefs mount -config` with its corresponding configuration file.

Re-creating the books database

Modify `bin/init.sh` and `.env` to use the path `./var/primary/fuse/books.db` instead of `./var/books.db`, remove `./var/books.db`, then run `foreman start`.

Once both replicas are up and running, run `./bin/init.sh` to create `books.db` in the primary FUSE directory. After a moment, a replica will be created in the secondary FUSE directory.

Use the `sqlite3` command to open the secondary replica. Can you `SELECT` from the books table? Can you `INSERT` or `DELETE`?

Testing

Modify `./bin/post.sh` to use the KrakenD port and the path `/api/books/`, then verify that you can run

```
./bin/post.sh ./share/book.json
```

successfully. Check that the change is reflected in both database replicas.

Documenting your results

In order to complete the exercise you will need to write a short report in PDF format documenting your results for each step.

At the beginning of your report, include:

- Your name
- Your partner's name (if you worked with someone else)
- The semester, course number, and section number
- The exercise number

For each item above:

- Clearly identify which step is being performed.
- Document performing the step itself, including any commands that were run and any code that was written or modified.
- Document the results of the step. Include diagrams, written descriptions of the results, or screenshots clearly demonstrating its success or failure.
- Analyze and comment on the results. Include references that you consulted, variations that you attempted, issues you encountered, and troubleshooting steps that you needed to perform.

Submit your work as a `.pdf` file through Canvas by the due date. If you work with another student, only one submission is required as long as it includes the names of both students.