Skipper High Level Overview

When Skipper starts, three processes are created: the webserver, the crawler, and the frontend.

webserver

Python Flask Webserver running at 127.0.0.1:500

crawler

Python script that continuously gathers info

frontend

Interactive Terminal Application written in Python and curses

If the webserver can't find an existing database, it will create one. The webserver then loads the user's kubernetes configuration file and finds all accessible clusters listed in the file.

database

Lightweight SQLite DB which serves as Skipper's single source of truth

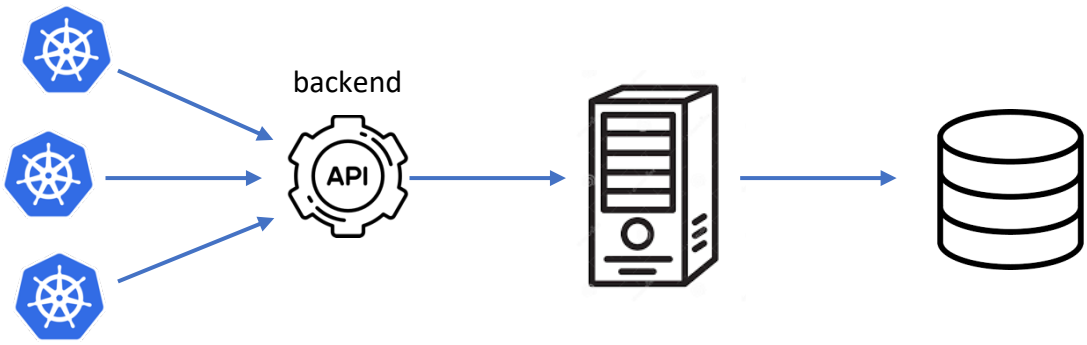
kube-config

File that Skipper reads to get access to Kubernetes clusters

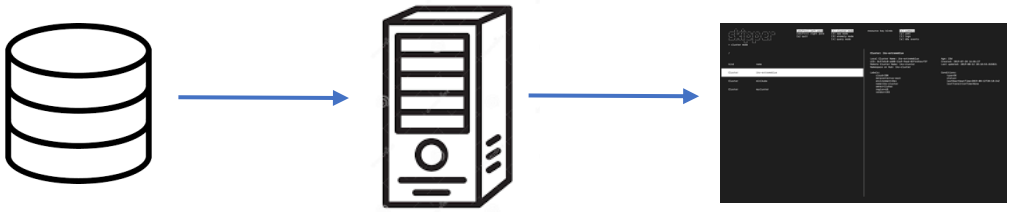
When the frontend needs information, it will make a request to the webserver.

GET /start/cluster
(get me all the info I need to start in cluster mode)

The webserver will query the database to see if the requested info is present. If an empty result is returned, the webserver will use Skipper's backend functions to get the info from the clusters and insert this info into the database.



In the end in all cases, the webserver will end up querying the database and sending the resulting information to the frontend, which updates the display with this information.



Meanwhile, in the background, the crawler is continuously gathering information from the clusters and using the endpoints given by the webserver to insert this info into the database.