Challenge Statement

This challenge is about creating a simple video storage server with REST APIs

Details

You are tasked to develop a simple video storage server with REST APIs, which should be implemented as follows:

- The APIs should be in accordance with the document sent separately. (This OpenAPI document only contains the minimum and may need to be added).
    - Upload a video file
    - Delete a video file
    - Download a video file
    - List uploaded files
- The server and other required services are configured as docker containers, and these will use a preconfigured docker network named **`restapi-challenge-network`**. This network is not connected to the Internet.
- The endpoints of the server are exposed to the host machine.

After you have written the server, please implement CLI interface for a user to interact with the server:

- The CLI named `video-store` should support the same four functions.
    - `video-store upload <path-to-the-video-file>`
    - `video-store delete <fileid>`
    - `video-store download <fileid>`
        - File will be stored in the current directory
    - `video-store list`
- We suggest using existing CLI frameworks to handle the needed boilerplate.

When working on this challenge, be sure to implement your solution as if you were in a production setting (think about how a production code repository is organized and set up and how you might distribute this software to a general user). Specifically, some things to think about include:

- Proving correctness of the code (and ensuring each module works the others)
- Error handling and bad inputs
- User friendliness of installation and setup

Additional features

After implementing the above, you want to implement one or more of the following features:

- Video searching by name, duration and so on
- Authentication and multi-user support
- Asynchronous video convert to webm (using ffmpeg etc.)
    - Create an API that shows the progress of the converting task.
    - Use same API to download converted data
- Data encryption and key management

Note

- These additional features should be implemented separately from the original tasks (in order to evaluate the original tasks independently).
- If you add or change APIs, include its OpenAPI document.

What to return back to us

1. The project folder and all its contents
2. A README.md file containing build and instructions
3. Please zip or tar everything in a directory named **yourfirst.lastname**/ and return via email

Additional Notes

The first part is meant to be the base level of measuring the candidate's ability and meant to be easily testable (please make sure that tests are included as this has been an immediate drop for candidates who have not bothered with it in the past).

As for the additional features, intent is for candidates to show additional expertise in another area outside of the main challenge. It isn't necessary to do all of them and would rather see someone try to do 1 really well rather than do all of them sloppily or "half-assed."