Coding Question: Optimize HTTP Server

Problem Statement:

You are tasked with implementing a simple HTTP server with the following requirements.

Implementation requirements

1. Set up an HTTP server in a language/framework of your choice.
2. The server should respond to incoming GET requests on the endpoint /data.
    a. You must accept 2 query params, **n:** file name**,** and **m:** line number**.**
3. If n and line both are provided, return the content of file /tmp/data/**n**.txt at line number **m**.
4. If only n is provided, return the contents of file /tmp/data/**n**.txt entirely.
5. Each file should be around 100MB in size, there will be more than 30 different files (eg; 1.txt, 2.txt ... 30.txt … n.txt).

Sample input and output:

Request: /data?n=1&m=30
Response: vyAF9kLDTIbqkv5R7hFqGDXaxezu3WMV5pcPd6RdudWMqMGJBQ9YLOoCQt

Request: /data?n=1&m=30
Response:
MSMJ53ZZt9BHPtgsuBwrSYeAG7N7HJW76aC85lajC2OCBU4oxkT6YDsVK9fxSHRCO
Cx7WP2Q9iXcFxiS1gjQaoVww5enIWX57Xj1cjxeAbvMALn37fuE0jv5SKtFqCZdLNdpcX5goGzf
DMtaN3H
oXEBnCjYAzYHl1p5X6YAQLNbqgjFoRoRpa84jDGXH4TNq2AqsUypnrYQOUlZwpp

Runtime requirements

1. Bundle everything inside a docker image (keep the docker file name as Dockerfile).
    a. Also, store the Dockerfile in the root project directory
2. Make sure the docker file is compatible with ARM architecture and x86.
3. Expose port 8080.
4. The Docker container should be allocated a maximum of 1500 MB RAM and 2000m/2 Core CPU.

Additional information:

1. You may use any libraries or frameworks that you find suitable.
2. Make sure your docker file is named.
3. Create data files with random text content in them for development purposes.
4. Provide any relevant documentation for your optimizations.
5. Go through all the requirements carefully and stick to them.