# Fileserver - Server Based File Storage

This is a fileserver based on remote filesystem. It provides an extensible design with hooks to integrate with any 3rd party fileserver including Managed service by cloud provider like AWS, Azure etc. This is a spring boot based project with cli as well. This project also includes a simple cli for file upload operations. Few other considerations in case of a web and distributed scenario are : ###Security & Privacy \* authentication/authorization- User authentication and RBAC control \* encryption - files could be encrypted in application level based on file sensitivity \* encryption keys:- Encryption keys must be kept securely preferably a managed Key-vault \* Rate-limit - Upload API should throttle to avoid any DDos attach \* signed url- there is no download url, but in case we implement that API, uri would be signed with an expiry certificates:-

### Performance && Scale:

* Caching - To improve performance of APIs like list files
* CDN:- Content delivery Network like AWS cloudfront for static downloads
* Disks replicated in multiple availability zones( datacenter) in a region
* Paritioning algorithm to split files into multiple disks keep Metadata into database

### Data Protection Compliance

* File should not be leaving regional boundary without user consent

### Availability ,Deployment & Disaster recovery:

Regional deployment with failover to different region

## Installation

prerequisite: Docker should be installed. It can be installed from https://www.docker.com/products/docker-desktop Run Following Command: - cd sampleFileServer - docker-compose -f docker\docker-compose.yml up

#### Build Locally

* Run ./mvnw install Or ./mvnw.cmd on Windows
* Run Server using ./fs-server for Linux/MacOS and ./fs-server.cmd on Windows ## CLI Usage Run CLI locally
* Run ./mvnw install -pl fs-cli for Linux/MacOs and ./mvnw.cmd install -pl fs-cli for Windows
* Run ./fs-store for Linux/MacOs and ./fs-store.cmd for Windows

fs-store scipt provides command to run jar produced in build step. Docker based image support can be added as well if need.

#### Note:

CLI also support pointing the server host URL by exporting environment variable FS\_STORE\_HOST. For Example export FS\_STORE\_HOST=http://localhost:8080 on Linux/MacOs. Similar support is available in Windows as well.

* FS\_STORE\_HOST=http://localhost:8080 => http/https schemes are supported currently and needs to be provided in URL
* Build docker CLI image require configuring network as well and passing ENV variable accordingly

Sample CLI Output

test@pixxs-MBP sampleFileServer % ./fs-store upload ~/zull-keystore.jks  
Uploading files..  
[1] File=/Users/test/zull-keystore.jks ...done  
 test@pixxs-MBP sampleFileServer % ./fs-store list  
Listing contents:  
[1] zull-keystore.jks  
test@pixxs-MBP sampleFileServer %

## Contributing

1. Fork it!
2. Create your feature branch: git checkout -b my-new-feature
3. Commit your changes: git commit -am 'Add some feature'
4. Push to the branch: git push origin my-new-feature
5. Submit a pull request :D

## Curl Examples

Upload file to Server:

curl -i -X POST -H "Content-Type: multipart/form-data" -F "file=@README.md" http://localhost:8080/v1/fileserver/files

List Files from Server

curl localhost:8080/v1/fileserver/files

## JDK installtion

We suggest Adopt SDK. Link: https://adoptopenjdk.net/

## Credits

Abhishek Ranjan

## License

NO CopyRights