**Phase1:**

Set Up backbone

* Create **S3 bucket(s)** with versioning & lifecycle rules.
* Define folder/file structure.
* Create **DynamoDB tables** for metadata (file ID, owner, tags, allowed roles, upload history).
* Write basic **Lambda functions** for:
* File upload (store metadata in DynamoDB).
* File download (retrieve from S3).
* Test with temporary hardcoded credentials (no Cognito yet).

**Phase 2:**

Secure the system with role and login

* Set up **Cognito user pool** (sign-up/sign-in).
* Define roles: **Admin, HR, Employee** (and any other needed).
* Map Cognito groups to **IAM roles**.
* Update Lambda functions to:
* Check logged-in user’s role before upload/download.
* Restrict pre-signed URL access by role.
* Add encryption (S3 server-side encryption, enforce HTTPS).

**Phase 3:**

Build copy-drive interface

* Set up **React/HTML frontend** (basic file manager UI).
* Integrate with Cognito login flow.
* Integrate API calls to Lambda (upload/download).
* UI features:
* File upload, download, delete
* Tagging & sorting
* Folder structure navigation
* View history/logs (from DynamoDB)

**Phase 4**

Finalise

* End-to-end **testing of role-based access** (Admin vs HR vs Employee).
* Test **expired pre-signed URLs** (access denied).
* Test lifecycle rules (old files auto-archived/deleted).
* Create **Security Architecture Document** (encryption, IAM roles, Cognito, data flow).