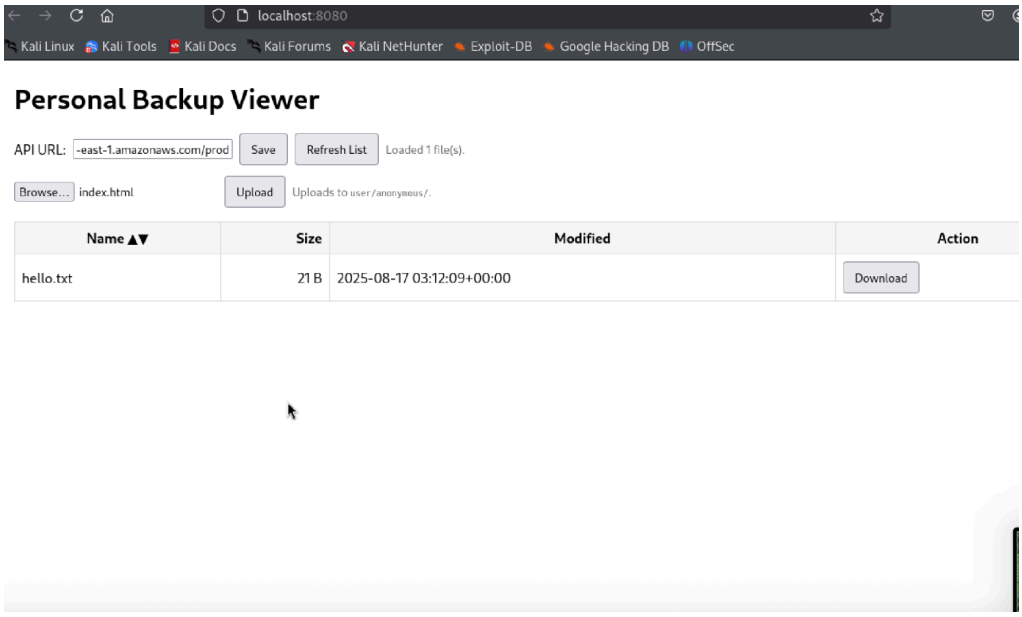
Personal Backup (Simulation) — Friendly One-Pager

Prepared by: *Tazahnae Matthews*

A tiny, safe way to upload, see, download, and delete files using a simple web page and a small AWS backend



What this is (plain English)

- A **web page** you open on your computer.

- A small **API** (one URL) that gives short-lived links to put/get files in an **S3 bucket**.
- It's a **simulation/demo** for learning and sharing. Keep it internal.

Little picture: You click buttons → page calls the API → API hands back a temporary link → file goes to/comes from S3.

You → Web Page → API Gateway → Lambda → S3 (private & encrypted)

Try it now (no AWS login needed)

1. **Open the page**
    - Double-click index.html in projects/personal-backup/web/,

or run:
python3 -m http.server -d /path/to/web 8080
then open http://localhost:8080/

    -

Paste API URL → click **Save** → **Refresh List**
https://cwjk3uxvnf.execute-api.us-east-1.amazonaws.com/prod

2.
3. **Use it**
    - **Upload**: pick a file → Upload
    - **Download**: click Download
    - **Delete**: removes the file
    - **Sort**: click the table headers (Name / Size / Modified)

Tip: If you see a "NetworkError", make sure the URL starts with **https://** and ends with **/prod**, then hard-refresh (Ctrl/Cmd-Shift-R).

How it works (kid-simple)

- S3 is a **locked toy box**.
- Lambda gives you a **magic key** (a presigned link) that works **once** and **expires in a few minutes**.
- The page just asks for a key and uses it.

Safety & scope

- Bucket is **private**, **encrypted**, and **versioned**.
- Links are **short-lived** and only for the one file.
- This demo is **open** (no login). Share internally; don't store secrets.

Next steps if we productize: add sign-in (Cognito), stricter IAM, and lifecycle rules to clean up old versions.

Common use cases (when this is handy)

- **Personal backups**: photos, scans, notes — version history lets you roll back.
- **Team drop folder**: a simple shared place to hand off files without AWS accounts.
- **Client intake**: let clients upload securely via a link; no login required (demo-safe).

- **Report delivery**: a lightweight "download center" for generated exports/artifacts.
- **DevOps demo**: a clean starter for Terraform + Lambda + API Gateway + S3.
- **DR stash**: back up critical configs/scripts to durable, versioned storage.
- **Classroom/lab**: teach presigned URLs, least privilege, and serverless basics.
- **CI/CD artifact sink**: scripts/pipelines can PUT via presigned URLs without long-lived creds.

Note: For sensitive data or external users, add authentication (e.g., Cognito), tighter IAM, and lifecycle rules.

Setup I completed (so you know the steps)

- Installed **Terraform** (distro had 1.6.3; upgraded to **1.12.2**).
- Installed **AWS CLI v2** (and **jq** for JSON testing).
- Created an IAM access key and configured profile **personal-backup** (`aws configure`).
- Set `AWS_PROFILE=personal-backup` and `AWS_REGION=us-east-1` (added to `~/.zshrc` so it persists in zsh).
- Deployed with Terraform: S3 (private, versioned, encrypted), Lambda, API Gateway.
- Enabled **CORS** on the API and added Lambda CORS headers.
- Gave Lambda S3 permissions: **Put/Get**, **ListBucket**, and (optionally) **DeleteObject**.
- Added S3 **lifecycle rules** to clean up old versions / incomplete uploads.
- Served the UI locally via `python3 -m http.server`.

Gotchas I hit (and how I fixed them)

- **"No configuration files"** when running terraform init/apply → I was in the wrong folder. Fixed by cd ~/projects/personal-backup/terraform (and creating main.tf).
- **HCL error: "Invalid single-argument block definition"** → broke one-line nested blocks into multi-line (e.g., the SSE rule under aws_s3_bucket_server_side_encryption_configuration).
- **Missing providers** → ran terraform init -upgrade to pull aws, archive, random.
- **Lambda code changes not deploying** → added source_code_hash = filebase64sha256(...) to the Lambda resource and used terraform apply -replace=aws_lambda_function.sign_url.
- **API 400 (op must be upload|download)**; paste that URL into the page.
