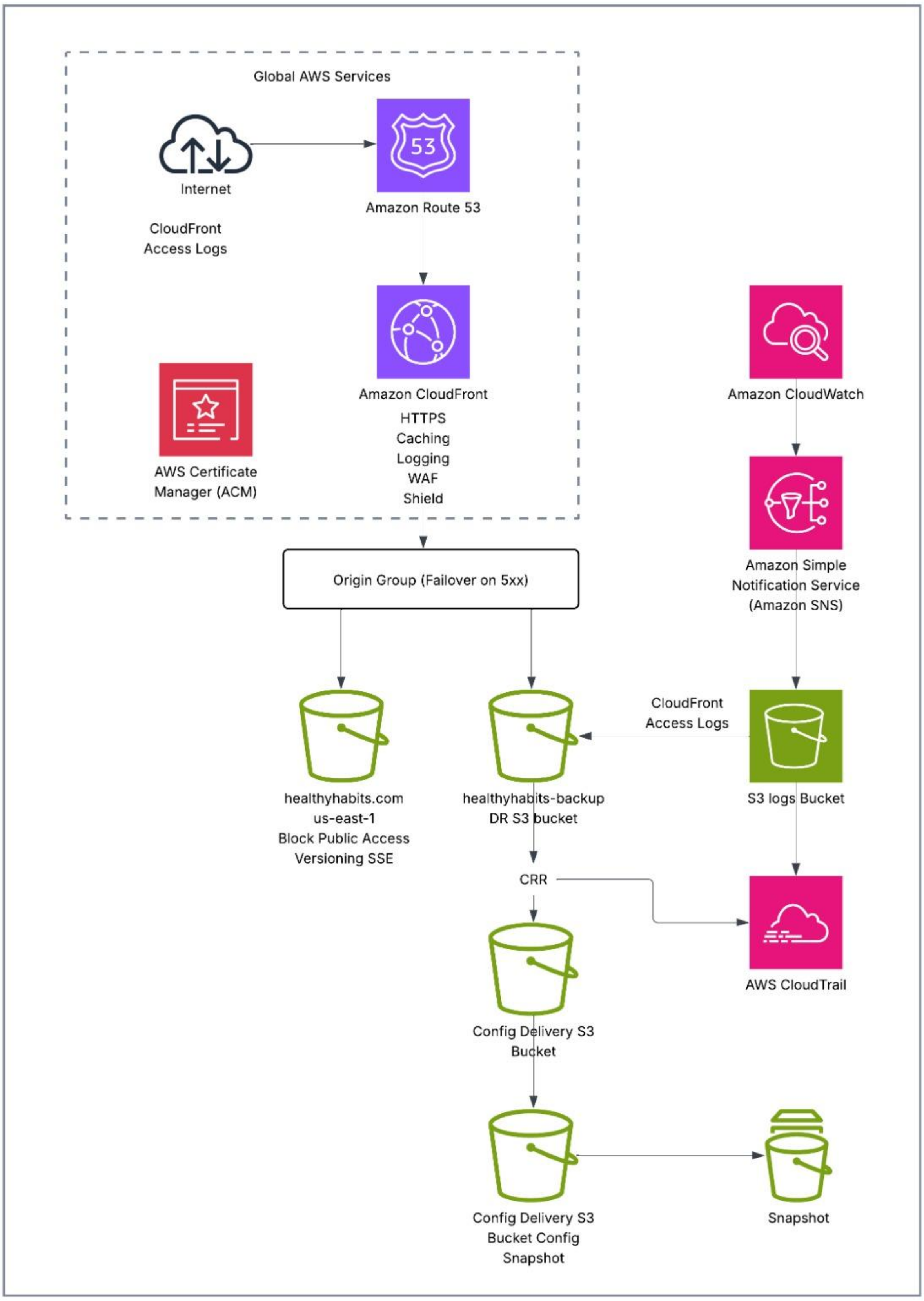
AWS Secure Static Website Hosting with Global Content Delivery, Monitoring, and Disaster Recovery

Project purpose: Host a production-ready static website on AWS using S3 + CloudFront with secure access, global performance, observability, and a simple disaster recovery strategy — implemented entirely via the AWS Management Console.



Stage 1 — Preparation (account, region, and basics)

1. Signing in to AWS Console: Create a AWS Root User Account
2. Pick two AWS Regions:
    - Primary region: The *origin S3 bucket* location (us-east-1).
    - DR region: for replication/backup (us-west-2).
3. Try to use Free Tier Resource
4. Or Enable Billing Alerts (optional): Create Cost budget to avoid surprise costs during demos.
5. Create an IAM user: For doing the rest of the operations.
    - *Secure the Root User Account and grant the only Permissions required to the IAM User (The Least Privilege Principle).*

Stage 2 — Create S3 bucket and upload website

1. Creating a S3 bucket
    - Bucket name: healthyhabits.com
    - Region: us-east-1
    - Keep all public access blocked
    - Enable versioning (for recovery).
    - Enable Server-side Encryption
    - Tags: Add: environment=prod, project=healthy-habits, website type=static.
2. Uploading the website files
    - Open the bucket → Upload → Add the files.
3. Enable S3 default encryption
    - Bucket → Properties → Default encryption → Enable SSE-S3 (or SSE-KMS).

Stage 3 — Create an ACM certificate (for HTTPS)

CloudFront requires a certificate in **us-east-1** for custom domain HTTPS.

1. ACM (us-east-1) → Request a certificate → Request a public certificate.
    - Add names: healthyhabits.com
    - Validation: DNS validation
    - Wait for validation

Stage 4 — Configuring CloudFront distribution

CloudFront will be the public face — it enforces HTTPS, WAF, caching, origin failover, and logging.

1. CloudFront → Create distribution → Web
2. Origin configuration
    - Origin domain: select the S3 bucket
    - Origin access: choose Origin Access Control (OAC)
    - Create an OAC:
        - Name: oac-healthyhabits-website.
        - Signing: Sign requests with OAC.
        - Protocol policy: HTTPS only.
    - Can use "Origin access identity (OAI)" but prefer OAC.
3. Default cache behavior
    - Viewer protocol policy: Enforce HTTPS
    - Allowed HTTP methods: GET, HEAD
    - Cache policy: Managed-CachingOptimized
    - Origin request policy: AllViewerExceptHostHeader (managed/default)
    - Enable Compress objects automatically.
4. Default root object: index.html.
5. SSL/TLS settings
    - Alternate domain names (CNAMEs): healthyhabits.com
    - Select ACM certificate created in us-east-1.
6. Logging
    - Enable Standard logs (free tier)
    - choose a separate logging S3 bucket
7. Enable WAF
8. Origin failover (for Disaster Recovery)
    - Add a second origin pointing to *DR S3 bucket*
    - Create an Origin Group with primary → failover to secondary.
    - Set failover criteria (HTTP 5xx)
9. Create the distribution and wait for deployment

Stage 5 — Set S3 bucket policy to allow CloudFront OAC only

1. S3 → select bucket → Permissions → Bucket policy → add a policy that allows only CloudFront OAC principal to GetObject.

    - ➢ Write the Policy:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AllowCloudFrontServicePrincipalReadOnly",
   "Effect": "Allow",
   "Principal": {
    "Service": "cloudfront.amazonaws.com"
   },
   "Action": "s3:GetObject",
   "Resource": "arn:aws:s3:::my-website-primary.example.com/*",
   "Condition": {
    "StringEquals": {
     "AWS:SourceArn": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE"
    }
   }
  }
 ]
}
```

2. Block public access remains enabled — direct S3 URL will be blocked.

Stage 6 — Domain (Route 53) and SSL validation

1. Route 53 → Hosted zones → creating a hosted zone for healthyhabits.com
2. Add DNS validation records for ACM: Add CNAME records by ACM to the hosted zone.
3. Create an Alias record to CloudFront
    - ○ Create Record → A (Alias) → Name- healthyhabits.com → Alias to CloudFront distribution → select distribution.
4. Wait for DNS propagation and ACM validation

Stage 7 — Disaster Recovery

S3 Cross-Region Replication + CloudFront failover (not free tier, optional)

Plan: keeping a replicated copy of objects in another region and configure CloudFront origin failover.

1. Create DR S3 bucket in DR region (healthyhabits-backup).
2. Enable versioning and encryption on DR bucket too.
3. Create IAM role for replication
4. S3 → Source bucket → Management → Replication rules
    - Add rule: replicate Entire bucket
    - Destination: us-west-2
    - Replicate existing objects — enable
    - Save and confirm
5. Verify replication: upload a new object to primary bucket and confirm it appears in DR bucket.
6. CloudFront origin failover
    - Create an Origin Group with Primary origin = healthyhabits.com and Secondary origin = healthyhabits-backup
    - Keep the status codes that trigger failover: HTTP 5xx
    - Attach the Origin Group to the default cache behavior as the origin.

Stage 8 — Security hardening

1. AWS Shield (Standard) is automatically enabled for CloudFront. (Free tier)
2. S3 permissions: Validate only required principals have access. Remove any public grants.
3. IAM best practices
    - Use IAM User for all the operations
    - Use IAM Role for S3 replication.
4. AWS Config
    - Enable AWS Config in both regions to record resource changes. This shows compliance and configuration history.
5. CloudTrail
    - CloudTrail → create a trail → apply to all regions → log delivery to a dedicated S3 bucket (enable encryption).
    - This captures API activity for auditing.

Stage 9 — Observability & Monitoring

1. CloudFront metrics (CloudWatch)
    - CloudFront automatically publishes metrics to Cloud Watch
    - CloudWatch → Metrics → CloudFront → create alarms: (not Free)
        - Alarm: 5xxErrorRate > 1% → SNS topic to notify by email
2. S3 metrics
    - S3 → your bucket → Management → Metrics → enable Request metrics
3. Set up SNS notifications
    - CloudWatch Alarms → create SNS topic → subscribe the email.

# Cost controls & cleanup

1. Cost controls
    - Budgets → Create monthly cost budget and alerts.
    - Turn off unused resources to avoid charges.
2. Cleanup steps
    - Delete CloudFront distribution (disable first).
    - Delete S3 buckets (empty first).
    - Delete ACM certificates if not used.
    - Remove Route 53 records if used a test domain.
    - Turn off AWS Config / CloudTrail (don’t need permanent logs).