Practical - 7: CloudFront Content Delivery Network (CDN)

Objective: To configure Amazon CloudFront with a private S3 origin, restrict direct S3 access, and demonstrate cache management via invalidation.

Phase 1: Prepare S3 Origin and Initial Setup

1. **Create Origin Bucket:** Navigate to the S3 console and create a new bucket (e.g., **pranav-cloudfront-pract7-origin**). Ensure **Block Public Access** is **ON** (the default and recommended security practice).
2. **Prepare Files:** Ensure you have the following files ready for upload:
    - index.html (Content: Version 1)
    - index-v2.html (Content: Version 2)
    - image.png
3. **Upload Files:** Upload index.html and image.png to the root of the **Origin Bucket**.
4. **Test Direct Access:** Attempt to access the S3 Object URL for index.html directly in a browser. This should fail with an **Access Denied** error because public access is blocked.

Phase 2: Create and Configure CloudFront Distribution

1. **Create Distribution:** Navigate to the CloudFront console and click **Create Distribution**.
2. **Specify Origin:**
    - **Origin Domain:** Select your S3 bucket (**pranav-cloudfront-pract7-origin**) from the dropdown menu.
    - **S3 Access:** Select **Allow private S3 access to CloudFront (Recommended)**.
    - **Origin Access Control (OAC):** Choose to **Create new OAC** to manage the secure connection between CloudFront and S3.
3. **Default Behavior and Settings:**
    - **Viewer Protocol Policy:** Select **Redirect HTTP to HTTPS**.
    - **Settings -> Default Root Object:** Set this to **index.html**.
    - **Distribution Name: pranav-cloudfront-pract7**.
4. **Create Distribution:** Click **Create Distribution**.
5. **Apply Bucket Policy (Critical Step):** After creation, CloudFront will display a generated **S3 Bucket Policy**.
    - Navigate back to your **Origin Bucket** in S3.
    - Go to **Permissions -> Bucket policy**.
    - Click **Edit**, paste the generated policy, and click **Save changes**.
    - *This policy grants the CloudFront service principal permission to read objects from the bucket*.

Phase 3: Validation and Caching Demonstration

1. **Wait for Deployment:** Wait until the CloudFront distribution status changes to **Deployed**.
2. **Test CDN Access:** Copy the **Distribution Domain Name** (e.g., d1lkbq...cloudfront.net) and paste it into a browser.
    - **Validation:** The website should load successfully, showing the **Version 1** content.
3. **Cache Miss:** Open your browser's developer tools (**Network** tab) and examine the response headers for the first request. The X-Cache header should show **Miss from cloudfront**.
4. **Cache Hit:** Refresh the page immediately. The subsequent request's X-Cache header should show **Hit from cloudfront**, proving the content is being served from the Edge Location cache.

Phase 4: Invalidation (Forcing Content Update)

1. **Update Origin Content (S3):**
    - In the S3 bucket, **Delete** the existing index.html.
    - **Rename** index-v2.html to **index.html**.
2. **Test Cache Status:** Refresh the CloudFront URL. It will still show the old **Version 1** content because the Edge Location has not been told to refresh the cache.
3. **Create Invalidation:**
    - In the CloudFront console, select the distribution, and go to the **Invalidations** tab.
    - Click **Create Invalidation**.
    - For **Object Paths**, enter **/*** (wildcard) to clear the entire cache.
    - Click **Create Invalidation** and wait for the status to change to **Completed**.
4. **Final Validation:** Refresh the website URL. It should now show the updated **Version 2 (UPDATED!)** content, confirming the cache was successfully cleared.