# Optimizing High Availability with CloudFront Origin Failover

## Overview

AWS CloudFront origin failover is a critical feature for ensuring high availability of web applications. It automatically switches to a secondary origin when the primary origin becomes unavailable, ensuring continuous content delivery to users.

## How Origin Failover Works

- **Origin Groups:** Configure at least two origins (primary and secondary)
- **Failover Criteria:** Define HTTP status codes that trigger failover
- **Automatic Switching:** CloudFront routes to secondary on primary failure
- **Primary Preference:** Always attempts primary origin first
- **Supported Methods:** Works with GET, HEAD, and OPTIONS requests

## Configuration Steps

- Ensure your CloudFront distribution has at least two origins
- Navigate to CloudFront console and select your distribution
- Create an origin group with primary and secondary origins
- Specify HTTP status codes for failover (400, 403, 404, 500, 502, 503, 504)
- Update cache behavior to use the origin group
- Test failover by simulating primary origin failure

## Common Use Cases

- **S3 Multi-Region:** S3 buckets in different regions for disaster recovery
- **EC2 Redundancy:** Multiple EC2 instances across availability zones
- **Hybrid Failover:** Combine with Lambda@Edge for advanced logic
- **Custom Error Pages:** Consistent user experience during failover

**High Availability:** Origin failover significantly enhances application reliability by ensuring content delivery even when primary origins fail. Combined with other AWS services, it provides comprehensive disaster recovery solutions.