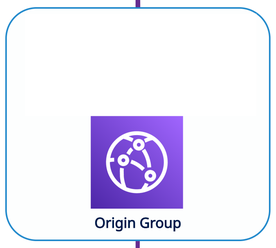
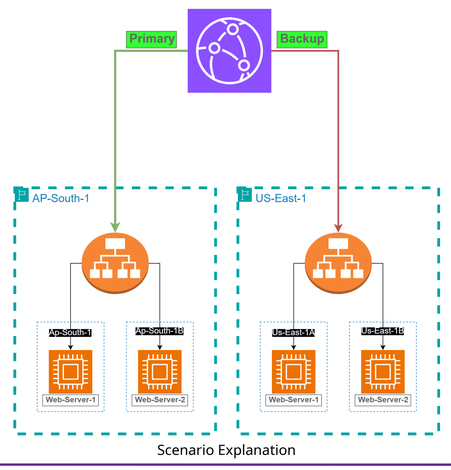
- 1 Introduction
  - 1 CloudFront Origin Group
    - 1 A CloudFront Origin Group is a collection of multiple origins
    - 2 It provides reliability for your content delivery
  - 2 Multiple Origins
    - 1 You can add multiple origins to the group
    - 2 These origins can be defined as
      - 1 Primary Origin
      - 2 Secondary Origin
  - 3 Primary Origin
    - 1 The primary origin is the main source for your content
    - 2 CloudFront sends all requests to the primary origin first.
  - 4 Secondary Origin
    - 1 The secondary origin acts as a backup
    - 2 If the primary origin fails, CloudFront switches to the secondary origin
  - 5 Automatic Failover
    - 1 CloudFront automatically checks the health of both origins
    - 2 It automatically switches to the secondary origin if the primary is unavailable
- 2 Lab Scenario - 1 Basic Scenario: EC2 and S3 Failover
  - 1 Scenario Explanation — In this scenario, the goal is to ensure continuous availability of your website even if the primary server (EC2 instance) goes down
  - 2 Why We Are Creating This Setup
    - 1 User Experience
      - 1 If the primary EC2 instance hosting your website goes down, we don't want users to see an error or have no response.
      - 2 Instead, we want to give users a friendly message that the site will be back soon, or provide them with a basic version of the website.
    - 2 Business Continuity — This setup ensures that your website remains accessible, even if it's in a limited capacity, which is crucial for maintaining trust with your users.
    - 3 Temporary Solution
      - 1 The S3 bucket will serve as a temporary, backup website
      - 2 It could display a "maintenance" message or provide basic information until the primary EC2 instance is back online.
  - 3 Hands-On Steps
    - 1 Step-1 Create an EC2 Instance and Configure it as web server
    - 2 Step-2 Set Up an S3 Bucket and configure static web hosting
    - 3 Step-3 Create a CloudFront Distribution
    - 4 Step-4 Setup Origin Group
    - 5 Step-5 Test the Setup



- 3 Lab Scenario - 2 Geographical Failover with Load Balancers
  - 1 Scenario Explanation



    - 1 In this advanced scenario, the goal is to ensure your website remains highly available and resilient, even in the event of a regional failure.
    - 2 This setup uses CloudFront with a geographically dispersed failover strategy.
  - 2 Why We Are Creating This Setup ?
    - 1 Global Reliability
      - 1 If the primary load balancer in the India region goes down, we want to ensure that users, regardless of their location, can still access the website.
      - 2 The failover to the N. Virginia region ensures that your site remains available even if an entire region experiences an outage
    - 2 Business Continuity on a Global Scale
      - 1 For businesses with a global audience, it's essential to have redundancy across multiple regions.
      - 2 This advanced scenario ensures that your business operations continue uninterrupted, even if one region goes down.
  - 3 Hands-On Steps
    - 1 Step-1 Create an EC2 Instances in both regions and Configure it as web servers
    - 2 Step-2
      - 1 Set up a load balancer in the India region and configure it to route traffic to your EC2 instances or web servers in that region
      - 2 Set up a load balancer in the N. Virginia region to route traffic to your backup resources in that region.
    - 3 Step-3 Create a CloudFront Distribution
    - 4 Step-4 Setup Origin Group
    - 5 Step-5 Test the Setup