

AWS[®]

Certified Developer

Official Study Guide

Associate (DVA-C01) Exam



```

"TargetPrefix": "examplebucket/",
"TargetGrants": [
  {
    "Grantee": {
      "Type": "AmazonCustomerByEmail",
      "EmailAddress": "user@example.com"
    },
    "Permission": "FULL_CONTROL"
  },
  {
    "Grantee": {
      "Type": "Group",
      "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
    },
    "Permission": "READ"
  }
]
}
}

```

Creating Custom Domain Name with Amazon Route 53

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to internet applications by translating names like `www.example.com` into the numeric IP addresses like `192.0.2.1` that computers use to connect to each other.

Amazon Route 53 is fully compliant with IPv6 as well.

You may not want to use the Amazon S3 endpoint such as `bucket-name.s3-website-region.amazonaws.com`. Instead, you may want a more user-friendly URL such as `myexamplewebsite.com`. To accomplish this, purchase a domain name with Amazon Route 53.



You can purchase your domain from another provider and then update the name servers to use Amazon Route 53.

Amazon Route 53 effectively connects user requests to infrastructure running in AWS—such as Amazon EC2 instances, Elastic Load Balancing (ELB) load balancers, or Amazon S3 buckets—and can route users to infrastructure outside of AWS. You can use Amazon

Route 53 to configure DNS health checks to route traffic to healthy endpoints or to monitor independently the health of your application and its endpoints. Amazon Route 53 Traffic Flow makes it easy for you to manage traffic globally through a variety of routing types, including latency-based routing, geolocation, geoproximity, and weighted round-robin, all of which can be combined with DNS failover to enable a variety of low-latency, fault-tolerant architectures.

Using Amazon Route 53 Traffic Flow's simple visual editor, you can easily manage how your end users are routed to your application's endpoints—whether in a single AWS Region or distributed around the globe. Amazon Route 53 also offers domain name registration. You can purchase and manage domain names such as `example.com`, and Amazon Route 53 will automatically configure DNS settings for your domains.

Speeding Up Content Delivery with Amazon CloudFront

Latency is an increasingly important aspect when you deliver web applications to the end user, as you always want your end user to have an efficient, low-latency experience on your website. Increased latency can result in both decreased customer satisfaction and decreased sales. One way to decrease latency is to use *Amazon CloudFront* to move your content closer to your end users. Amazon CloudFront has two delivery methods to deliver content. The first is a web distribution, and this is for storing of `.html`, `.css`, and graphic files. Amazon CloudFront also provides the ability to have an RTMP distribution, which speeds up distribution of your streaming media files using Adobe Flash Media Server's RTMP protocol. An RTMP distribution allows an end user to begin playing a media file before the file has finished downloading from a CloudFront edge location.

To use Amazon CloudFront with your Amazon S3 static website, perform these tasks:

1. Choose a delivery method.
In the example, Amazon S3 is used to store a static web page; thus, you will be using the Web delivery method. However, as mentioned previously, you could also use RTMP for streaming media files.
2. Specify the cache behavior. A cache behavior lets you configure a variety of CloudFront functionality for a given URL path pattern for files on your website.
3. Choose the distribution settings and network that you want to use. For example, you can use all edge locations or only U.S., Canada, and Europe locations.

Amazon CloudFront enables you to cache your data to minimize redundant data-retrieval operations. Amazon CloudFront reduces the number of requests to which your origin server must respond directly. This reduces the load on your origin server and reduces latency because more objects are served from Amazon CloudFront edge locations, which are closer to your users.

The Amazon S3 bucket pushes the first request to Amazon CloudFront's cache. The second, third, and n^{th} requests pull from the Amazon CloudFront's cache at a lower latency and cost, as shown in Figure 13.1.