FILE SYNCHRONIZATION SERVICE

Amazon EC2

Auto Scaling

Elastic Load Balancing

Amazon **Route 53**

Amazon **DynamoDB**

Amazon **S3**

AWS **STS**

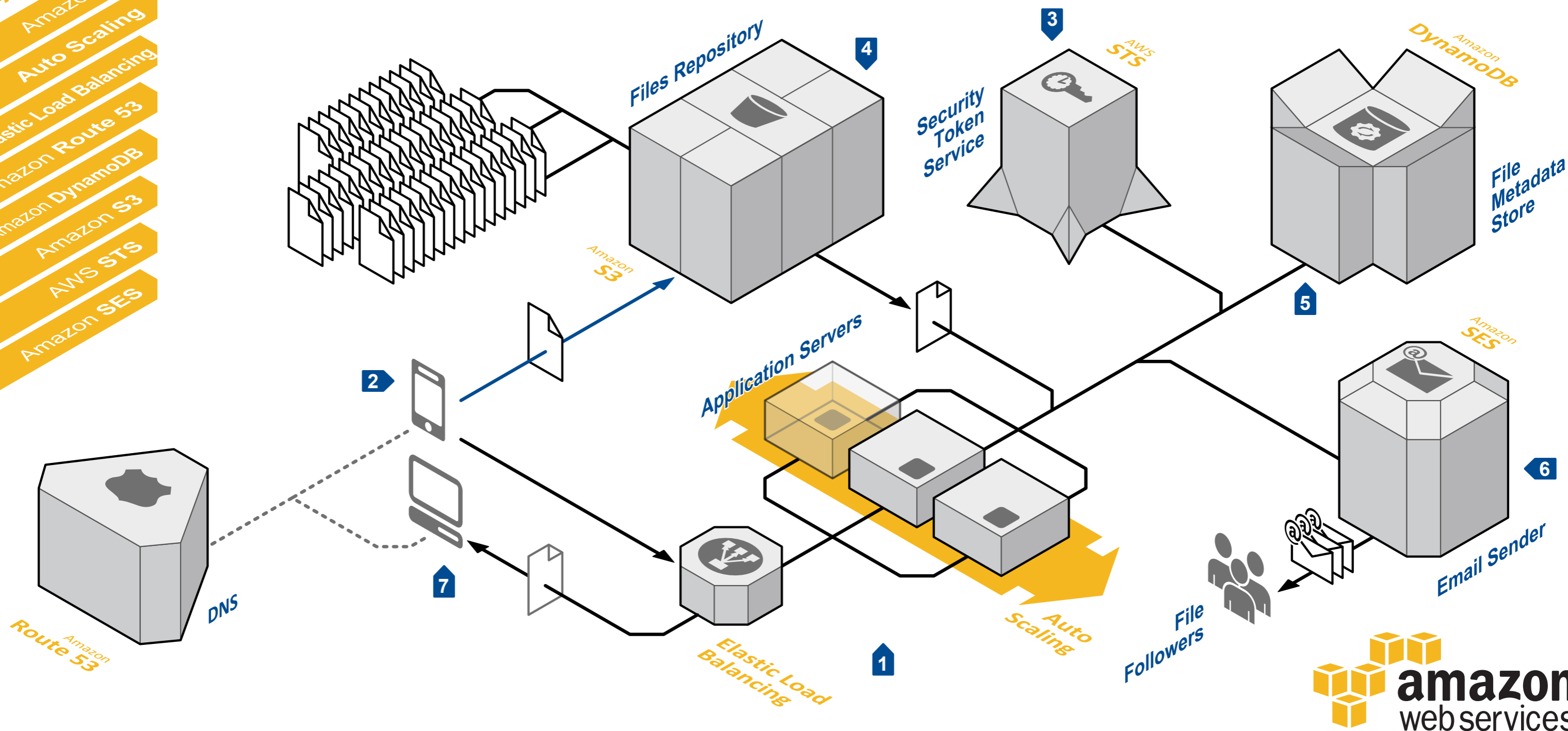Amazon **SES**

Given the straightforward, stateless client-server architecture in which web services are viewed as resources and can be identified by their URLs, development teams are free to create file sharing and syncing applications for their departments, for enterprises, or for consumers directly.

This diagram represents the core architecture of a scalable and cost-effective file sharing and synchronization platform, using Amazon Web Services.

Files Repository

Amazon S3

Security Token Service

AWS STS

Amazon DynamoDB

File Metadata Store

Application Servers

Amazon SES

Email Sender

DNS

Amazon Route 53

Elastic Load Balancing

Auto Scaling

File Followers

amazon web services

System Overview

1 The file synchronization service endpoint consists of an **Elastic Load Balancer** distributing incoming requests to a group of application servers hosted on **Amazon Elastic Compute Cloud** (Amazon EC2) instances. An **Auto Scaling** group automatically adjusts the number of **Amazon EC2** instances depending on the application needs.

2 To upload a file, a client first needs to request the permission to the service and get a security token.

3 After checking the user's identity, application servers get a temporary credential from **AWS Security Token Service** (STS). This credential allows users to upload files.

4 Users upload files into **Amazon Simple Storage Service** (Amazon S3), a highly durable storage infrastructure designed for mission-critical and primary data storage. **Amazon S3** makes it easy to store and retrieve any amount of data, at any time. Large files can be uploaded by the same client using multiple concurrent threads to maximize bandwidth usage.

5 File metadata, version information, and unique identifiers are stored by the application servers on an **Amazon DynamoDB** table. As the number of files to maintain in the application grows, **Amazon DynamoDB** tables can store and retrieve any amount of data, and serve

any level of traffic.

6 File change notifications can be sent via email to users following the resource with **Amazon Simple Email Service** (Amazon SES), an easy-to-use, cost-effective email solution.

7 Other clients sharing the same files will query the service endpoint to check if newer versions are available. This query compares the list of local files checksums with the checksums listed in an **Amazon DynamoDB** table. If the query finds newer files, they can be retrieved from **Amazon S3** and sent to the client application.