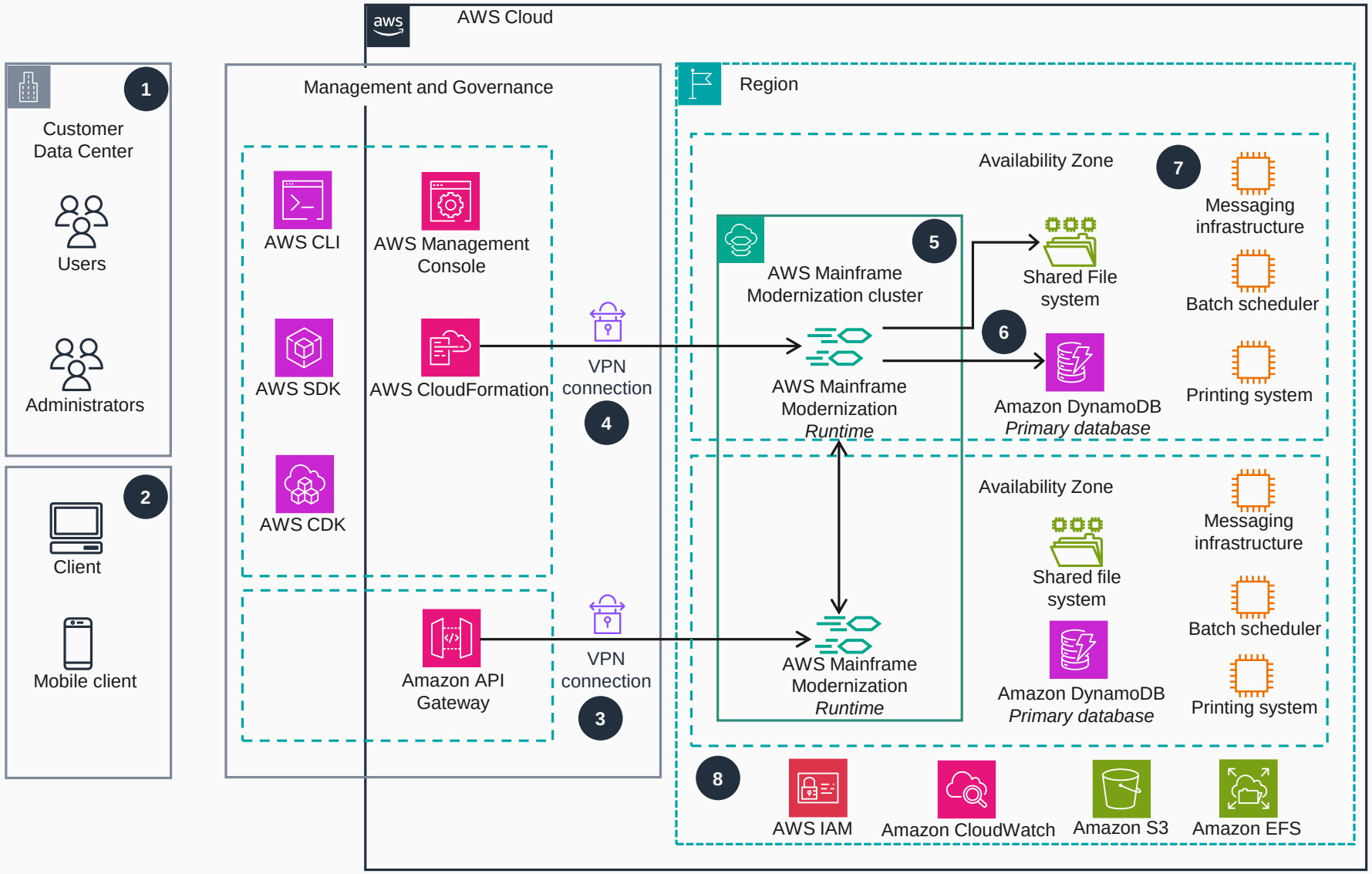
Guidance for Operating Mainframe Applications in the Cloud with AWS Mainframe Modernization

This architecture diagram shows how to operate modernized mainframe applications at scale using the various management interfaces with the AWS Mainframe Modernization service.



1. Users access the service using the **AWS Command Line Interface (AWS CLI)** or the AWS Management Console (users are prompted to sign in or create an account). Users can also deploy migrated applications using **AWS CloudFormation** templates or **AWS Cloud Development Kit (AWS CDK)** scripts.

2. The network traffic between these AWS services and the **AWS Mainframe Modernization** service is always encrypted.

3. With a web browser or mobile device, end users access the modernized mainframe application on AWS through calls to the **AWS Mainframe Modernization** APIs.

4. **AWS Mainframe Modernization** uses HTTPS for API endpoints, which are also configured by default.

5. An application in **AWS Mainframe Modernization** contains a migrated mainframe workload. These applications can be deployed in multiple Availability Zones. It is analogous to a workload on the mainframe and is associated with a runtime environment. Users can add batch files and data sets to applications and monitor applications as they run. Users create **AWS Mainframe Modernization** applications for each migrated workload.

6. Both refactored and replatformed applications use a database and/or a file system. Users must create, configure, and manage the database according to specific requirements for each runtime engine. **Amazon FSx** or **Amazon Elastic File System (Amazon EFS)** can be used for storing application data on files.

7. Third-party solutions for messaging, scheduling, and printing integrate with **AWS Mainframe Modernization** using its management interfaces.

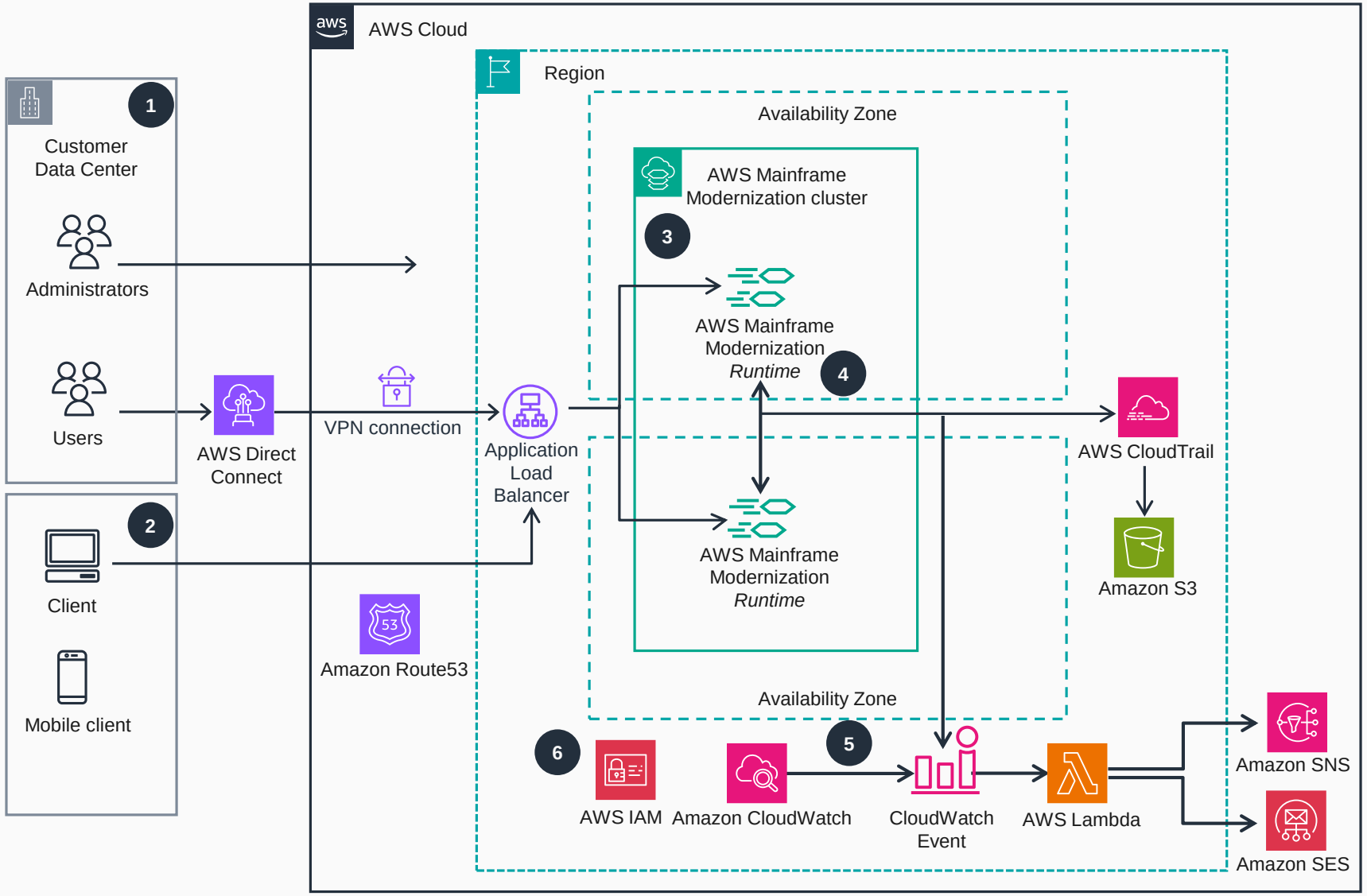
8. **AWS CloudTrail** enables operational and risk auditing, governance, and compliance of a user's AWS account. Actions taken by a user, role, or AWS service are recorded as events in **CloudTrail**. **AWS Identity and Access Management** (IAM) will return 'allow' or 'deny' based on standard **IAM** policy evaluations. **Amazon Simple Storage Service (Amazon S3)** is used for storing binaries, application build artifacts, configuration files, console, application, and database logs.

Reviewed for technical accuracy April 17, 2024

AWS Reference Architecture

Guidance for Operating Mainframe Applications in the Cloud with AWS Mainframe Modernization

This architecture diagram shows how applications modernized using AWS Mainframe Modernization have native integrations with AWS Cloud management and governance services.



1. Users can connect to AWS services over the network with **AWS Direct Connect**.
2. Clients can use **Amazon Route 53** to do name resolution and use **Elastic Load Balancing** to set up an Application Load Balancer that distributes traffic across multiple targets.
3. The **AWS Mainframe Modernization** service achieves cloud management by providing a managed runtime that takes care of various aspects of centralized logging, monitoring, billing, and licensing for operating in the cloud. Modernized mainframe applications on AWS benefit from cloud governance by having preventative and detective controls, account management, and auditing available on AWS.
4. Events in the user's AWS account, including events for **AWS Mainframe Modernization**, create trails in **CloudTrail**. **CloudTrail** can deliver these trails as log files to an **Amazon S3** bucket.
5. **Amazon CloudWatch** monitors the user's AWS resources, including the applications on **AWS Mainframe Modernization,** in real-time. Users can collect and track metrics by creating rules in **CloudWatch Events** to invoke **AWS Lambda** functions when a specified metric reaches a threshold that users specify. These notifications are received using various services, like **Amazon Simple Notification Service** (Amazon SNS) or **Amazon Simple Email Service** (Amazon SES).
6. The **AWS Mainframe Modernization** service supports **IAM** identity-based policies.

Reviewed for technical accuracy April 17, 2024

AWS Reference Architecture