CapitalOne Migration Case Study

Introduction:

Capital One Financial Corporation is an American bank holding company specializing in credit cards, auto loans, banking, and savings accounts. The company is headquartered in McLean, Virginia. By December 31, 2023, Capital One reported total assets of approximately $478.5 billion.

Capital One has been recognized for its commitment to community service, it was featured in PEOPLE's 2024 list of "100 Companies That Care".

Capital One officially became a 100% cloud-managed company in **2020**.
The bank had exited all **8** of its **on-premises** Data Centers.

Pre-migration:

- Centralized, physical infrastructure (8 data centers)
- Manual provisioning and long release cycles
- Limited personalization and data analytics
- Reactive incident management and scalability issues

The Migration Process:

Assessment – Prioritized workloads for migration

Refactor (80%) – Rebuilt cloud-native applications

Replatform – Leveraged AWS managed services

Rehost (Lift-and-Shift) – Used for lower priority apps

Upskilling – Trained 11,000+ engineers in AWS & Agile practices

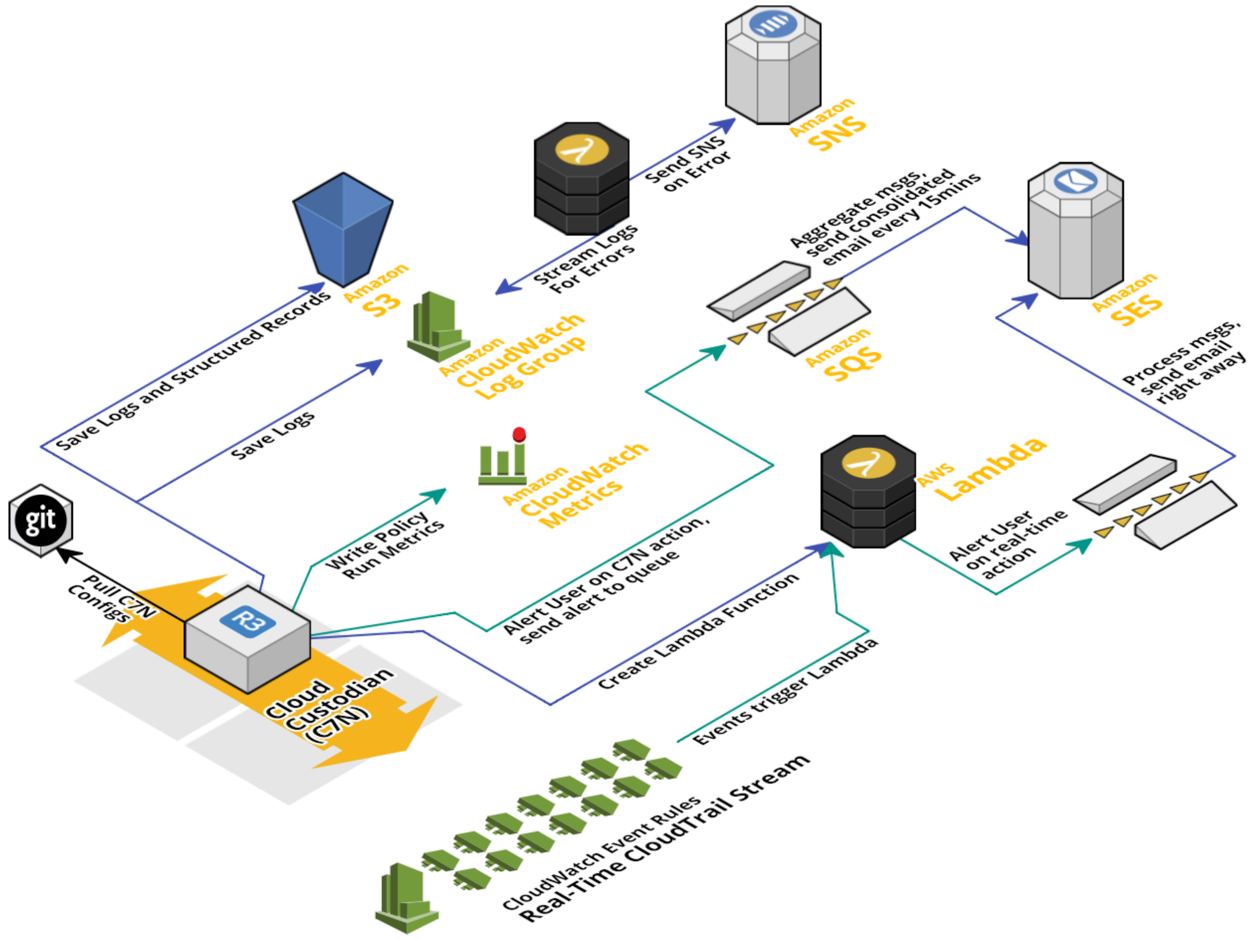
The Migration Hurdle (Challenges):

Challenge	Solution
💾 Growing storage costs	Lifecycle policies + Glacier tiering
⚙️ Legacy system complexity	AWS DMS for DB migration + refactor apps
🧠 Organizational shift	DevOps & agile training across 11,000+ engineers
🐢 App performance issues	Used **load balancing**, and tuned **Tomcat**

AWS Services in Use:

1. Amazon EC2	Compute power for scalable, secure VMs where containers/serverless were not viable
2. AWS Lambda	Serverless functions to handle backend logic with zero infra management
3. Amazon S3	Primary object storage; used for both static assets and big data
4. S3 Glacier & Deep Archive	Cold storage to cut costs on infrequently accessed data
5. Amazon RDS (PostgreSQL, MySQL)	Managed relational databases for internal apps
6. Amazon Aurora	High-performance, scalable relational DB for customer-facing apps
7. AWS DynamoDB	NoSQL storage for real-time personalization and session data
8. AWS CloudFormation	Automated infrastructure provisioning (IaC)
9. AWS Database Migration Service (DMS)	Migrate on-prem databases to RDS/Aurora with minimal downtime
10. Amazon CloudWatch	Monitoring and logging across all cloud infrastructure
11. AWS CodePipeline + CodeDeploy	CI/CD pipelines for agile deployment practices
12. AWS KMS (Key Management Service)	Encrypted data at rest and in transit (compliance)
13. AWS IAM + AWS Organizations	Managed access control and multi-account security governance
14. Amazon SageMaker	Building and deploying ML models for fraud detection and personalization
15. Amazon ECS/EKS	Container orchestration (for microservices and scalable APIs)
16. AWS WAF + AWS Shield	Protection from DDoS and common attack patterns

Architecture Diagram: AWS Policy Enforcement at Scale with Cloud Custodian


Amazon SNS
Send SNS on Error
Stream Logs For Errors
Amazon S3
Save Logs and Structured Records
Amazon CloudWatch Log Group
Save Logs
Aggregate msgs, send consolidated email every 15mins
Amazon SES
Amazon SQS
Process msgs, send email right away
Amazon CloudWatch Metrics
Write Policy Run Metrics
git
Pull C7N Configs
R3
Cloud Custodian (C7N)
Alert User on C7N action, send alert to queue
AWS Lambda
Alert User on real-time action
Create Lambda Function
Events trigger Lambda
CloudWatch Event Rules
Real-Time CloudTrail Stream

Architecture Explanation:

---

1) Policy Engine with Cloud Custodian:

Cloud Custodian (C7N) is the heart of this system.

- It checks your AWS resources (such as EC2 instances, S3 buckets, etc.) against a set of rules (policies) to ensure they follow your company's or security team's guidelines.
- These rules are defined in configuration files that are stored in a Git repository.
- Allows Versioning and Management of policies using code.

---

2) Logging and Recording Events:

After Cloud Custodian checks the resources, it creates records and logs.

Amazon S3: Logs and detailed records are saved here.

Amazon CloudWatch Logs: Collects the logs generated by Cloud Custodian so you can review them in a central place.

Amazon CloudWatch Metrics:

- In addition to logs, Cloud Custodian sends numbers (metrics) that represent how often a policy is triggered or how many errors there are.
- These metrics can be used to build dashboards or to create alerts if something unusual happens.

---

3) Detecting Issues and Sending Alerts:

When something goes wrong, the system must detect and notify the right people.

Monitoring Logs for Errors:

- CloudWatch Logs isn't just for storing messages;
- It is also monitored to detect error messages or other signs that a resource is not complying with your policies.

Amazon SNS (Simple Notification Service):

- When an error or important event is detected, CloudWatch sends a message to SNS.
- SNS is like a messaging hub that can deliver these alerts to different services or team members. It can send messages to:

    - **Amazon SQS (Simple Queue Service):** For message queuing.
    - **Amazon SES (Simple Email Service):** For sending out email alerts.

---

4) Handling Notifications and Email Alerts

Alerts can be delivered in two different ways:

- **Immediate Email Alerts:**
  - A part of the system is designed to send out emails right away if a critical issue is detected. This is usually managed by a **Lambda function** that triggers **SES (**Simple Email Service**)**.
- **Consolidated (Batched) Email Alerts:**
  - Other notifications may be collected in a queue (using SQS) and then processed together every 15 minutes.
  - This helps to avoid overwhelming the team with too many individual emails and provides a summary of events for a short period.

---

5) Automated Actions with AWS Lambda

AWS Lambda:

- Lambda functions are small programs that get triggered in response to events.
- They can automatically take actions like sending emails, starting remediation tasks (correcting issues), or integrating with other systems.

Triggers for Lambda:

- **CloudWatch Event Rules:** These are rules that watch for specific patterns in CloudTrail or CloudWatch events.
- **CloudTrail Events:** CloudTrail records all API calls made in your AWS account (like when resources are created or modified). When specific events occur, they trigger Lambda functions to act right away.

---

6) Real-Time Monitoring and Feedback Loop

CloudTrail Real-Time Event Stream:

- CloudTrail captures details of every API call made in your AWS environment, which is crucial for security and compliance auditing.
- These real-time events are monitored by CloudWatch rules that trigger actions immediately if something important happens.

- **Feedback Loop:**
  - The overall system creates a loop:
    1. **Detect:** Cloud Custodian checks your resources.
    2. **Log:** Information is saved to S3 and CloudWatch.
    3. **Alert:** Errors or issues trigger notifications through SNS and SES.
    4. **Act:** Lambda functions automatically perform any required actions.
  - This loop ensures continuous monitoring, immediate notification, and swift remediation.

Post-Migration State – What Changed?

1. Entirely cloud-native infrastructure on AWS.
2. Serverless architecture (AWS Lambda, S3, EC2, RDS, etc.).
3. Scalable, automated CI/CD pipelines.
4. Agile teams with full-stack ownership.
5. Personalized banking using real-time data and ML.

Cost Optimization & Savings

1. **Exact cost not publicly disclosed**
2. **Cost Optimization Steps: (Approach Used)**
    1. Shut down all 8 data centers (massive infra savings)
    2. Used Glacier & S3 tiering to cut storage costs
    3. Serverless & on-demand infra minimized provisioning waste

Benefits After Migration

1. Dev environment setup : Improvement From 3 months → Minutes.
2. Disaster Recovery : Improved by 70%.
3. Transaction errors : Reduced by 50%
4. Incident resolution time : Cut by 50%
5. Sustainability : Saved energy equal to powering 650,000 LED bulbs/year; recycled 103 tons of materials.