Financial Services Industry Lens

AWS Well-Architected Framework

June 2020

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Contents

Introduction 1
General Design Principles 2
Scenarios 3
- Financial Data 3
- Regulatory Reporting 4
- Artificial Intelligence and Machine Learning 5
- Grid Computing 6
- Open Banking 8
- User Engagement 9

Pillars of the Well-Architected Framework 10
- Operational Excellence Pillar 10
- Security Pillar 16
- Reliability Pillar 41
- Performance Efficiency Pillar 50
- Cost Optimization Pillar 52

Conclusion 53
Contributors 53
Document Revisions 54

Abstract

This document describes the **Financial Services Industry Lens** for the AWS Well-Architected Framework. The document describes general design principles, as well as specific best practices and guidance for the five pillars of the Well-Architected Framework.

Introduction

The financial services industry includes financial services firms, independent software vendors (ISVs), market utilities, and infrastructures that supply essential services to countries around the world. The system provides the main mechanism for paying for goods, services, and financial assets; intermediates between savers and borrowers—channeling savings into investment; and insures against and disperses risk.

The AWS Well-Architected Framework helps you understand the pros and cons of decisions you make while building systems on AWS. By using the Framework, you learn architectural best practices for designing and operating reliable, secure, efficient, and cost-effective systems in the cloud. The Framework provides a way for you to consistently measure your architectures against best practices and identify areas for improvement. We believe that having well architected systems greatly increases your security, reliability, and the likelihood of business success.

In this “Lens” we focus on how to design, deploy, and architect financial services industry (FSI) workloads that promote the resiliency, security, and operational performance in line with risk and control objectives that you define, including those to meet the regulatory and compliance requirements of supervisory authorities.

All customers should start with the best practices and questions outlined in the AWS Well-Architected Framework whitepaper. This document provides additional best practices for financial services institutions.

The Financial Services Industry Lens specifies best practices for security, data privacy, and resiliency that are intended to address requirements of financial institutions based on our experience working with financial institutions around the world. It provides guidance on guardrails for technology teams to implement to confidently use AWS to build and deploy applications. This Lens provides guidance on building transparency and auditability into your AWS environment. It provides suggestions for controls to help you expedite adoption of new services into your environment.

This document is intended for those in technology roles, such as chief technology officers (CTOs), architects, developers, engineers, and operations team members, as well as individuals in risk, compliance, and audit functions.

General Design Principles

The Well-Architected Framework identifies a set of four general design principles to facilitate good design in the cloud for financial services workloads.

1. **Documented operational planning**—To define your cloud-operating model, you must work with internal consumers and stakeholders to set a common goal and strategic direction. Many organizations have adopted the "Three Lines of Defense" model to improve effectiveness of risk management:

   - At the first line of defense, operational managers are responsible for executing risk and control procedures on a day-to-day basis.
   - The second line establishes various risk management and compliance functions to help build and/or monitor the first line-of-defense controls.
   - As the third line of defense, internal auditors provide the governing body and senior management with comprehensive assurance based on the highest level of independence and objectivity within the organization.

   Establishing clear roles and responsibilities across the three lines of defense is vital to developing an effective operating model for regulated cloud adoption.

2. **Automated infrastructure and application deployment**—Automation enables you to execute and innovate quickly and scale security, compliance, and governance activities across your cloud environments. Financial services institutions that invest in automated infrastructure and application deployment are able to accelerate the rate of deployments and more easily embed security and governance best practices into their software development lifecycle.

3. **Security by design—**Financial services institutions must consider a Security by Design (SbD) approach to implement architectures that are pre-tested from a security perspective. SbD helps implement the control objectives, security baselines, security configurations, and audit capabilities for applications running on AWS. Standardized, automated, prescriptive, and repeatable design templates help accelerate the deployment of common use cases as well as meet security standards (and ease the evidence requirements for audit) across multiple workloads. For example, to protect customer data and mitigate the risk of data disclosure or alteration of sensitive information by unauthorized parties, financial institutions need to employ encryption and carefully manage access to encryption keys. SbD allows you to turn on encryption for data at rest, in transit, and if necessary, at the application level by default.

4. **Automated Governance**—Human working with runbooks and checklists often lead to delays and inaccurate results. Automated governance provides a fast, definitive governance check for applications deployment at scale. Governance at scale will typically address the following components:
    - Account Management—Automate account provisioning and maintain good security when hundreds of users and business units are requesting cloud-based resources.
    - Budget and Cost Management—Enforce and monitor budgets across many accounts, workloads, and users.
    - Security and Compliance Automation—Manage security, risk, and compliance at scale to ensure that the organization maintains compliance, while executing against business objectives.

Scenarios

The following are common scenarios that influence the design and architecture of your financial services workloads on AWS. Each scenario includes the common drivers for the design and a reference architecture.

Financial Data

Access to financial data for workloads running on the cloud is a key component for the operations of financial services institutions. Examples of these datasets include real-time and historical market data, alternative data such as consumer movement, and buying decisions that can be analyzed for insight.

Financial data architectures supporting these use cases share the following characteristics:

- They have strict requirements around user entitlements and data redistribution.
- They have low latency requirements that vary depending on how the market data is used (for example, trade decision vs. post trade analytics), and can vary from seconds to sub-millisecond.
- They use a reliable network connectivity for market data providers and exchanges.

Reference Architecture

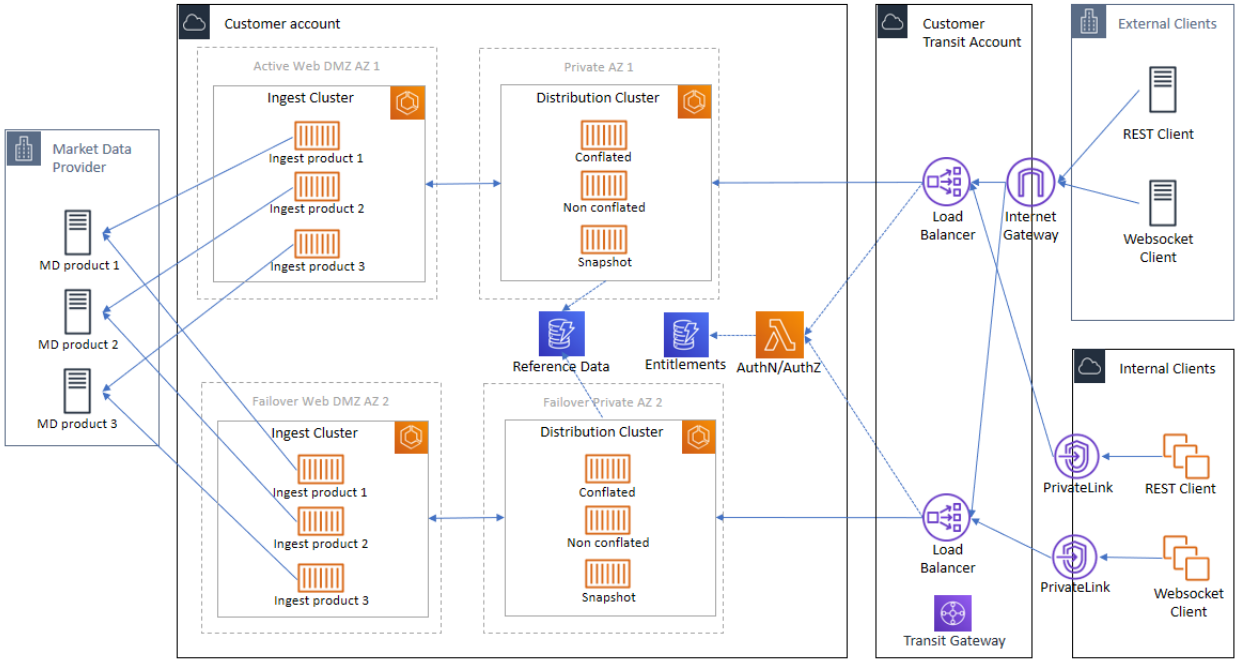

Figure 1: Reference architecture for a market data distribution platform within an enterprise

Regulatory Reporting

Every financial institution deals with volumes of information for regulatory reporting, and new regulations such as the European Union (EU) Markets in Financial Instruments Directive II (MiFID II) and U.S. Securities and Exchange Commission (SEC) Rule 613 (Consolidated Audit Trail) include reporting requirements. Static legacy infrastructure and inefficient reporting processes can make reporting costly and prevent customers from responding quickly to regulatory changes. Building a reporting data lake on AWS and leveraging the rich set of services can address many of the issues that complicate regulatory reporting (such as data residing in disconnected silos and distributed ETL processes). After customers integrate reporting data into a consistent dataset, they can use that data to gain additional insights through advanced analytics and machine learning.

Financial services data lake architectures supporting these use cases share the following characteristics:

- They implement data quality, integrity, and lineage into the ingest and processing pipelines.
- They require that data is encrypted at rest and in transit.
- They mask or tokenize personally Identifiable Information (PII) data to meet regulatory requirements (e.g. EU General Data Protection Regulation).
- They use Data Catalog with fine-grained access control and entitlements.

Reference Architecture

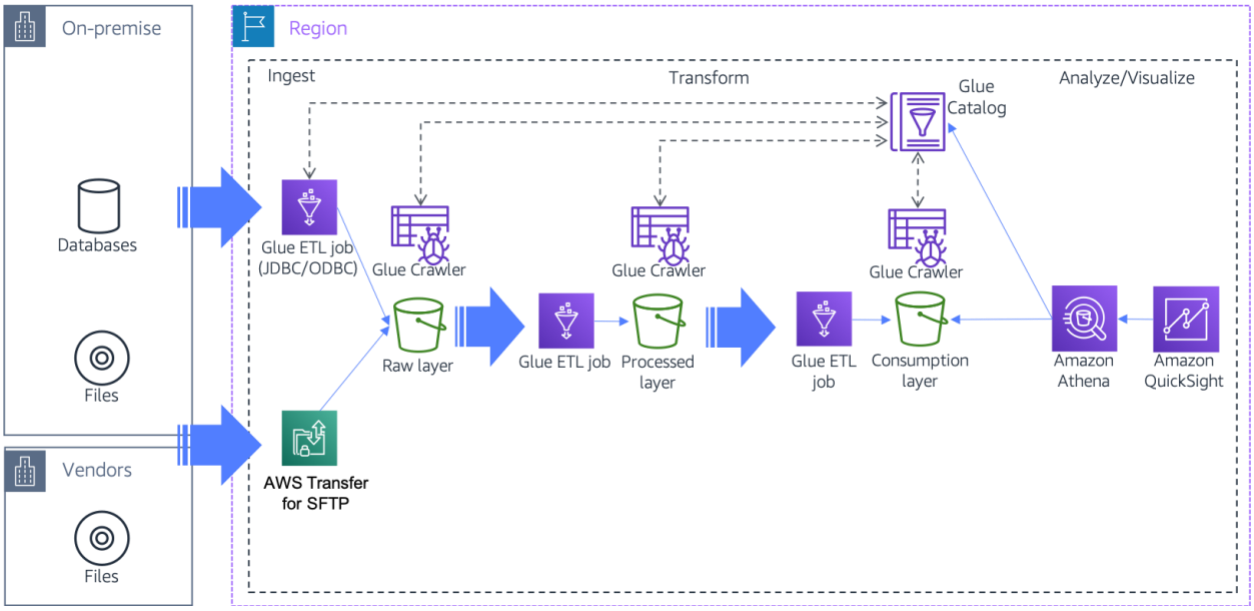

Figure 2: Reference architecture for a Financial Services Industry Data lake

Artificial Intelligence and Machine Learning

Financial institutions have been experimenting with artificial intelligence and machine learning (AI/ML) technologies for years, but the integration of these technologies into day-to-day operations has advanced slowly due to a lack of in-house data science expertise and insufficient experience manipulating large datasets. AWS provides a set of tools that make AI/ML readily accessible to any organization. Financial institutions are using these tools to enhance customer interactions through chatbots, improve surveillance, gather trading ideas from unstructured data, and customize product offerings, among many other use cases.

Financial services AI/ML architectures supporting these use cases share the following characteristics:

- They have a secure architecture to protect code and model artifacts.
- They have self-service capabilities for model development and training environments with pre-defined security configurations.
- They use a CI/CD pipeline integrated with change control systems for model deployment.
- They automate end to end evidence capture of the entire model development lifecycle across development, training, and deployment.

Reference Architecture

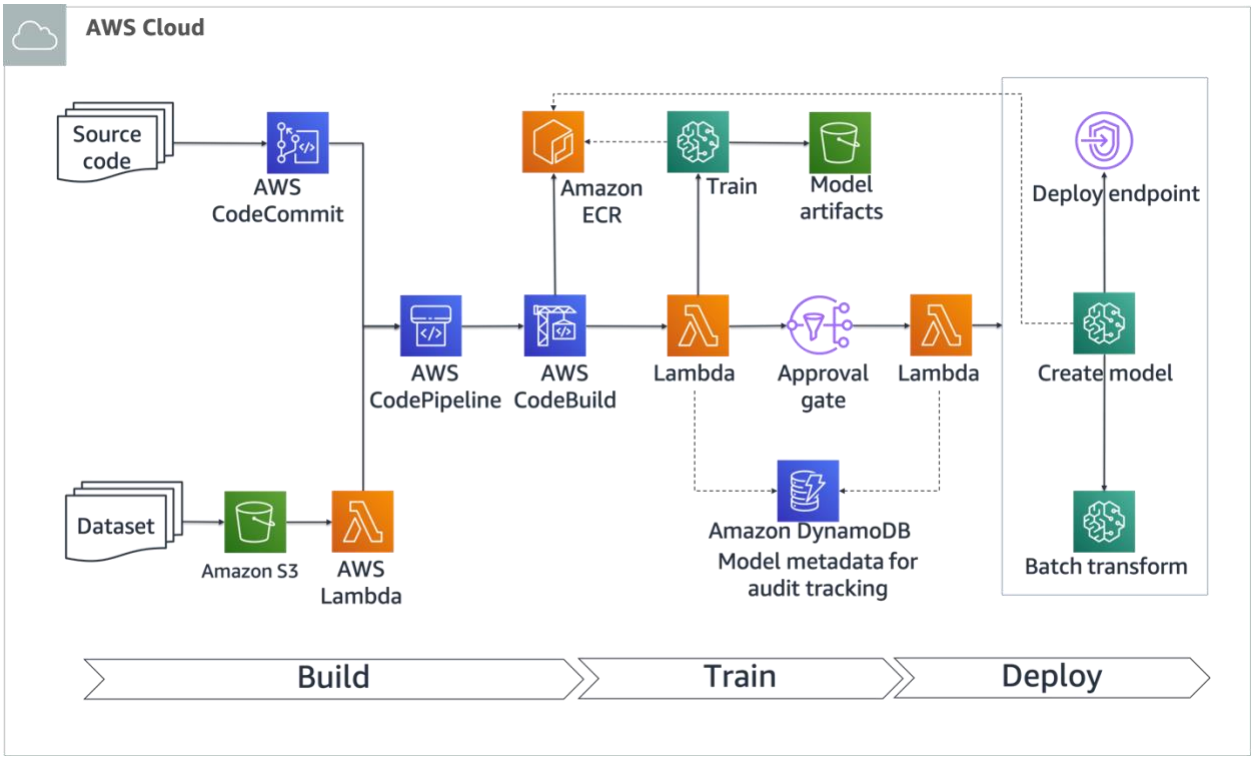

Figure 3: Reference architecture for an AI/ML pipeline

Grid Computing

Financial simulations are essential to the operations of all types of financial institutions in order to understand and manage risk, fully comprehend capital positions, conduct what-if testing, and make informed investment and pricing decisions. To run these simulations, financial institutions rely on clusters of computing resources (grids). However, a number of factors can make calculations more demanding, such as the proliferation of different varieties of relevant data; regulatory requirements to perform

higher levels of stress testing; and the increasing complexity of back-testing of new quantitative trading strategies and more complex products. To address their grid-computing needs, financial institutions are using AWS for faster processing, lower total costs, and greater accessibility.

Financial services high performance computing (HPC) architectures supporting these use cases share the following characteristics:

- They have the ability to mix and match different compute types (CPU, GPU, FPGA).
- They leverage Spot Instances to significantly reduce grid cost.
- They leverage Amazon Simple Storage Service (S3), Amazon Elastic File System (EFS) or Amazon FSx for persistent storage.

Reference Architecture

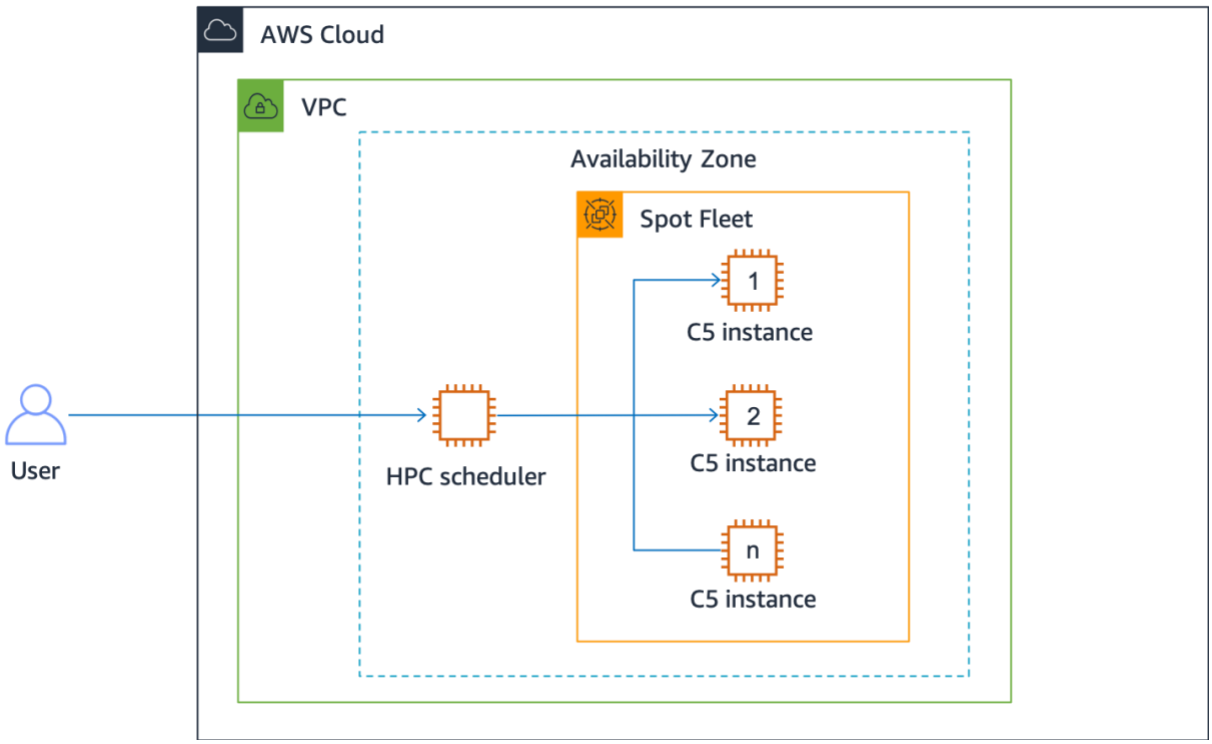

Figure 4: Reference architecture for an HPC Grid using Spot

Open Banking

In Open Banking, banks use Advanced Programming Interfaces (APIs) to securely share their customer data with third-party developers and service providers — allowing automated and secure access to the functionality of their core banking platform. Banks are building open banking platforms in response to new regulations and customer demands. Banks building their open APIs choose AWS because of the scalability, cost effectiveness, and the services AWS offers for analyzing large volumes of new data.

Open Banking architectures supporting these use cases share the following characteristics:

- They use an OAuth 2.0 authorization standard.
- They have an API driven infrastructure and elastic and scalable environment.
- They provide instant or near-instant access to customer account data.
- They have tamper-resistant logging and audit capabilities.

Reference Architecture

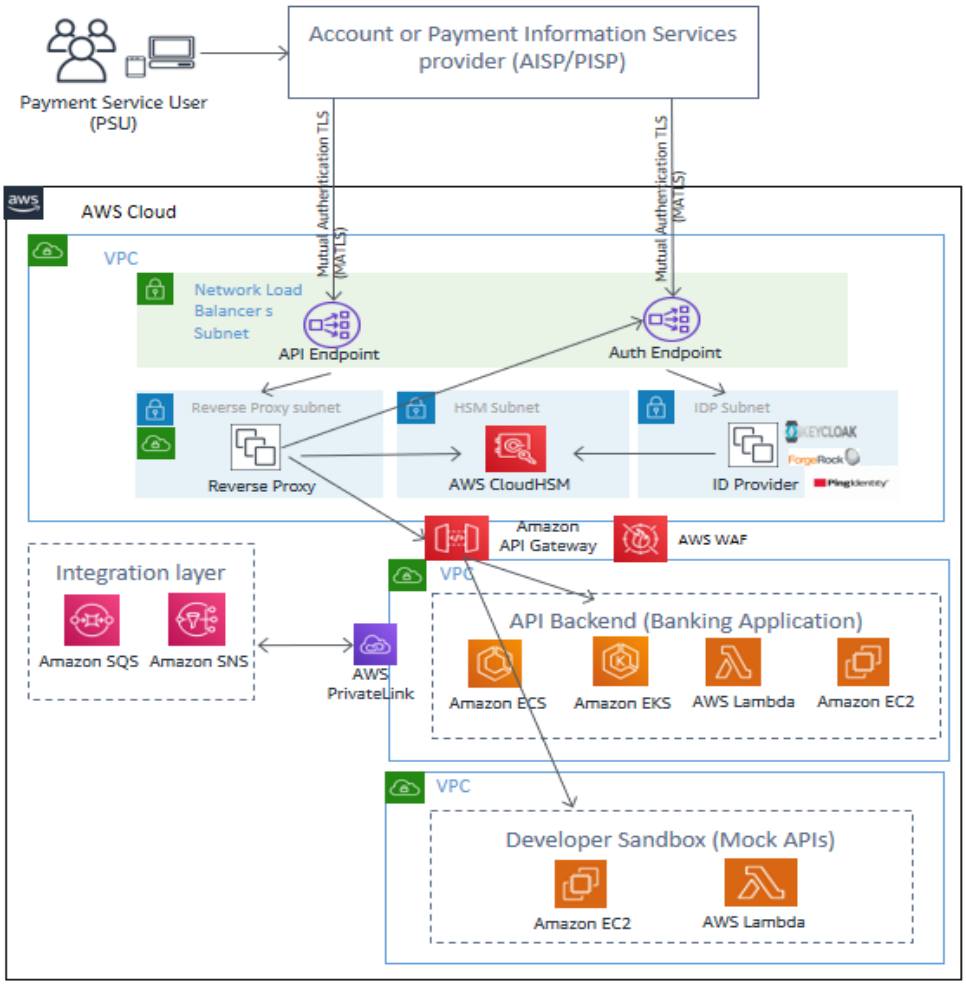

Figure 5: Reference architecture for Open Banking

User Engagement

Financial institutions are increasingly investing in their own customer-facing channels: mobile applications, web portals, call center agents and chatbots, advisors/brokers — all to enhance the overall customer experience.

Financial services user engagement architectures supporting these use cases share the following characteristics:

- They use high volumes of real-time data ingestion from public and private sources.
- Require different data protection considerations based on data classification.

- Employ event-driven architectures to leverage on-demand scalability and pay-per-use model.
- Includes real-time and archival data flows.

Reference Architecture

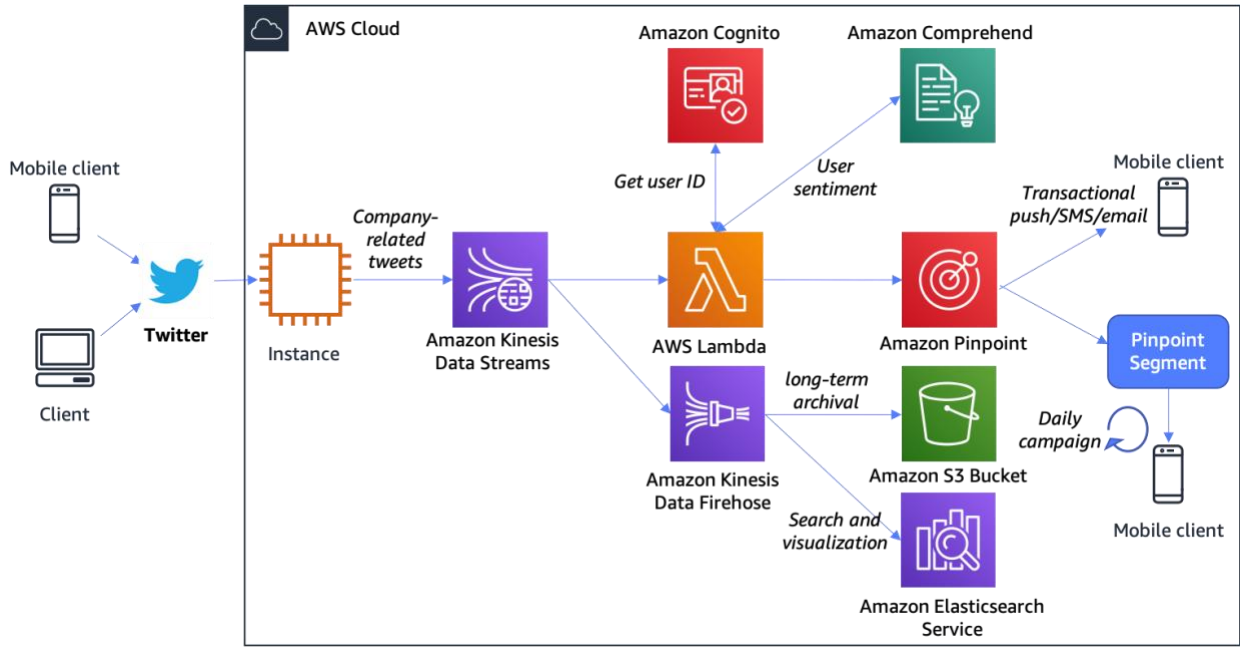

Figure 6: Reference architecture for real-time user engagement based on social sentiment

Pillars of the Well-Architected Framework

Operational Excellence Pillar

The **operational excellence** pillar includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures.

Financial institutions must focus on operational excellence, including the preventative measures and capabilities related to people, processes, and operating models. Focusing on this area allows financial institutions to adapt and recover quickly if things go wrong.

FSIOPS1: Have you defined risk and compliance roles for the cloud?

Define roles and responsibilities across risk functions

As explained in the General Design Principles section above, financial institutions typically adopt a Three Lines of Defense model to improve effectiveness of risk management. The 2nd and 3rd lines of defense must have the appropriate skills and training necessary to understand the risks involved in the delivery of business services using cloud – services owned and managed by the 1st line. Clear roles and responsibilities need to be established both within and across the three lines of defense functions to ensure the effectiveness and auditability of cloud operating model. These roles and responsibilities must be re-assessed at regular intervals to ensure that the governance model continues to be efficient and effective.

Engage with your risk management and internal audit functions to implement a process for the approval of cloud risk controls

Significant changes in technology, including migrating to the cloud necessitate a refreshed assessment of potential risks, and validation that the control environment cannot just mitigate identified risks, but also evidence their effectiveness. Engagement with the risk and internal audit functions will help ensure required governance obligations are met as cloud usage increases. This engagement needs to include training and education by the 1st line, to the 2nd and 3rd on the controls, technology, and processes that have been implemented to secure and operate the cloud environment. This process can contain a regular review cadence for new controls so the 1st line can evolve their implementations as needed so best practices for new threats can be adopted quickly, but safely.

Implement a process for adopting appropriate risk appetites

Failures can happen at any time. The appropriate risk authority within the firm (for example, Board of Directors or Chief Risk Officers or Business Risk Officers) needs to evaluate the criticality of a business process (and the underlying workloads that support that process) and specify the level of availability that the firm requires for such process. This must take into consideration the potential impact that a disruption of that process has on the firm, the customers, the financial infrastructure, and the cost of operating the workload in a high availability mode vs business agility and innovation. Working backwards from these risk appetites allows you to drive the operational priorities and the resiliency design choices of cloud workloads supporting business services in a prioritized manner. Setting clear risk appetites enables effective risk management and governance.

FSIOPS2: Have you completed an operational risk assessment?

Ensure that you understand the Shared Responsibility Model and how it applies to Services and Workloads you run in the cloud

In connection with your use of cloud, you must understand how the AWS Shared Responsibility Model affects your control environment. For example, certain controls may be the responsibility of AWS but certain controls remain the responsibility of the financial services institution. Review the AWS shared responsibility model and map AWS responsibilities and customer responsibilities according to each AWS service you use and your control environment. For those controls that are AWS's responsibility, you can use AWS Artifact to access AWS audit reports and review the implementation and operating effectiveness of AWS security controls.

Develop an enterprise cloud risk plan

A good approach is to map the interactions between business consumers of cloud services, and the internal stakeholders that shape this consumption, including risk and control considerations, Integrating across the three lines of defense functions, and ensuring they have the resources and training needed to satisfy their mandates for operating and protecting your business in the cloud while you strive to achieve your strategic goals is key. This integration can be achieved by carrying out a risk-based assessment of your operating model, and is especially effective when complemented with a review of decision-making processes and authority to ensure they are cloud-appropriate. As requirements are translated into controls, attention must be given to the strength of the controls (to ensure the identified risks will be mitigated), and the ability for the control design and performance to be evidenced (to facilitate independent assessment by internal risk management and audit functions). Focus on control design will ensure key control requirements are incorporated into the design from the start.

FSIOPS3: Have you assessed your specific workload against regulatory needs?

Implement a process for the review of applicable compliance and regulatory requirements for your workload

Financial services institutions must ensure that they are aware of all applicable regulatory and compliance obligations for their use of cloud and that they take

appropriate steps to meet those obligations. As part of your strategy, review your migration plan and control frameworks with the relevant internal stakeholders responsible for compliance to identify any compliance requirements, including legal and regulatory requirements that apply to your use of cloud. Note that designing a workload to meet specific technical requirements may only be one aspect of compliance so ensure you conduct a comprehensive regulatory and compliance review. This process must include both initial design and planning, as well as pre-production readiness activities.

Ensure that you also have a process to monitor evolving changes to compliance and regulatory obligations. The AWS Compliance Center is one resource you can use to learn about some key cloud-related regulatory requirements that may impact your use of cloud.

FSIOPS4: How do you assess your ability to operate a workload in the cloud?

Implement change management process for cloud

Cloud IT change management processes facilitate changes to IT systems in order to minimize risks to production environment while adhering to policies, audit and risk controls. It is not uncommon, especially within financial services institutions, to see a gated change management process often requiring a review by external change advisory boards which can take days or even weeks. As organizations take advantage of configuration management, infrastructure as code, automated testing and validation, and continuous integration and delivery, they can implement lightweight approval processes that are tightly integrated into CI/CD pipeline tools.

By leveraging automation to detect and reject bad changes, many of the manual approval steps can be fully automated with a higher degree of confidence. Even in highly regulated industries such as financial services where external reviews are required, reviews should still be integrated with the overall pipeline—even if they are manual steps initially. All tests, validations, approvals and rejections must be documented as part of the pipeline deployment. This will enable auditors to have a complete record of all applied changes including which environment the test and validations were run and who (and when) approved each change.

Financial services institutions must develop cloud capabilities in layers, producing approved, reusable artifacts at each layer such as Golden Amazon Machine Images

(AMIs), CloudFormation Templates, Service Catalog Products, etc. Artifacts at foundational layers must go through a change control process to ensure they comply with enterprise guidelines which can then be repurposed as building blocks by the rest of the organization. As the organization builds higher-level applications on a foundation of certified artifacts, the change control process will be expedited as it only needs to focus on the higher-level artifacts, significantly accelerating change while minimizing risk and ensuring compliance. Over time organizations develop capabilities to administer most of the changes in automated fashion with only a subset of changes that require manual intervention.

A good change management process enables the delivery of business value while balancing risk against business value. It should do so in a way that maximizes productivity and minimizes wasted effort or cost for all participants in the process. Automation, integration, and deployment tools in the Cloud allow businesses to make small, frequent changes that reduce risk and deliver business value at an increased rate. Please review the Change Management in the Cloud whitepaper for additional best practices.

Implement infrastructure as code

The benefit of cloud and infrastructure as code is the ability to build and tear down entire environments programmatically and automatically. If architected with resiliency in mind, a recovery environment can be implemented in minutes using AWS CloudFormation templates or AWS Systems Manager automation. Automation is critical for maintaining high availability and fast recovery.

AWS offers a wide breadth of automation tools to accomplish resiliency objectives. AWS Systems Manager helps automate complete runbooks that are used during the recovery of an application during a disaster. You can sequence a complete set of operations to automatically execute on the detection of an event. With Systems Manager automation documents, you can manage these runbooks similar to the way you manage code. You can version them and update them along with every release. This helps keep your recovery plan in sync with released code and updates to infrastructure.

Prevent configuration drift

Drift between primary and secondary sites can lead to failure in recovery during a disaster event. Financial services institutions should monitor changes to application infrastructure by using AWS CloudTrail and AWS Config. These services provide the capability to monitor activity within your AWS account, including actions taken through

the AWS Management Console, AWS SDKs, command line tools, and other AWS services. Once detected, you can automate the reactive action by defining workflows using Amazon CloudWatch Events integration.

Implementation of code-based management practices across your infrastructure, applications, and operational procedures enable high degrees of version control, testing, validation, and mitigation of human error and configuration drift that are necessary to limit the introduction of errors into your environment, and reduces the Mean Time to Recover (MTTR).

FSIOPS5: How do you understand the health of your operations?

Using enhanced monitoring in the cloud

High availability for workloads that support critical functions requires the ability to detect failures and quickly recover from them. You can understand the operational state of your workloads by defining, collecting, and analyzing metrics in the cloud that can be incorporated into your operating model. These metrics are emitted by your code, workloads, and user activity, and need to be collected in a centralized queryable system that can be used to visualize and examine real-world performance data. This is important for diagnosing issues that are often not clear from looking at just at application logs, Amazon CloudWatch, or system logs in isolation.

Monitor cloud provider events

Financial institutions should use the AWS Personal Health Dashboard, which provides alerts and remediation guidance when AWS is experiencing events that may impact workloads. The dashboard displays relevant and timely information to help manage events in progress, and provides proactive notification to help plan for scheduled activities. With Personal Health Dashboard, alerts are triggered by changes in the health of the AWS resources used in your applications, giving you event visibility and guidance to help quickly diagnose and resolve issues. Enterprise support and business support customers who have access to the AWS Health API can use this API to integrate the information from Personal Health Dashboard into the centralized monitoring system and define a consistent and comprehensive alerting mechanism.

FSIOPS6: Have you developed a continuous improvement model?

Test, model, and simulate scenarios before rollout

One of the best practices to determine if you have addressed your risk with appropriate controls is to actually run scenarios against your cloud control framework and operational procedures. Once your risk and control program is established, financial institutions should continuously asses and optimize their operational processes. Regular "game days" for workloads deployed on AWS can help build your team's muscle memory and validate that all operational procedures are effective in supporting your recovery objectives. We recommend designing game days to test your risk appetite and include severe, but plausible scenarios.

Conduct post-event operational reviews

Post-event operational reviews should be conducted after an incident. After troubleshooting and execution of repair procedures, follow-up documentation and actions should be assigned. A good post-event review results in a list of practical actions that address each of the issues that allowed the threat actor to succeed. These actions should minimize the impact of the event and teach the wider enterprise how to prevent, detect, and respond to a similar event in the future. For significant events, a Correction of Errors (COE) documents should be composed such that the root cause is captured and preventative actions may be taken for the future. Implementation of the preventative measures should be measured in future operations meetings.

Security Pillar

The **security** pillar includes the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies.

Customers, counterparties, and regulators expect financial institutions to maintain a strong cybersecurity posture. Given the shared responsibility model between AWS and customers, additional attention needs to be directed to understanding which aspects AWS covers, and which aspects the customer covers.

AWS Identity and Access Management (IAM)

FSISEC1: How do you ensure that AWS IAM Roles are compliant with the principle of least privilege?

FSISEC2: How do you monitor the use of elevated credentials, such as administrative accounts, and guard against privilege escalation?

Design functional IAM Roles based on the principle of least privilege.
Create roles which have the minimum set of policies, scoped down with applicable actions, resources and conditions. For example, you can create a role for all data scientists within your organization and allow them access to only AWS data analytics services with ReadOnlyAccess to specific buckets/keys.

You can also share resources in one AWS account with users in a different AWS account by delegating access across AWS accounts using AWS IAM Roles. To allow users from one account to access resources in another, create a Role that defines who can access it and what permissions it grants to users that switch to it. You can limit the Role's permissions to only what the Role requires to perform its function, aligned with the principle of least privilege.

Review IAM Policies
IAM Policies are powerful and subtle, so it's important to study and understand the permissions that are granted by each policy. For more information, read the Tips for Reviewing IAM Policies.

Review permissions using service last accessed data

Perform periodic reviews of your IAM Roles using Service Last Accessed Data. You can view a report about the last time that an IAM entity (user or role) attempted to access a service. You can then use that information to refine your policies to allow access to only the services that are in use. You can generate a report for each type of resource in IAM. For more information, read the Viewing Service Last Accessed Data process documentation.

Perform role reviews and delete unused roles
Perform periodic reviews of your IAM roles and delete roles that are not in use. Before you delete a role, review its recent service-level activity by viewing service last accessed data report.

Mitigate privilege escalation
Privilege escalation refers to the ability of a bad actor to use stealthy permissions to elevate permission levels and compromise security. Privilege escalation can result from misusing a number of non-administrator or non-full access permissions; for example, IAM:CreatePolicyVersion. This permission allows a user without administrator privileges to create a new custom permission and set it as the default version for a policy, even without access to the IAM:SetDefaultPolicyVersion permission.

To avoid scenarios like this, pay attention to the following permissions:

- IAM:AddUserToGroup
- IAM:AttachRolePolicy
- IAM:AttachUserPolicy
- IAM:CreateAccessKey
- IAM:CreateLoginProfile
- IAM:CreatePolicyVersion
- IAM:CreateRole
- IAM:CreateUser
- IAM:DeleteRole
- IAM:DeleteRolePermissionsBoundary
- IAM:DeleteRolePolicy
- IAM:DeleteUserPermissionsBoundary
- IAM:DeleteUserPolicy
- IAM:DetachRolePolicy
- IAM:PassRole
- IAM:PutRolePermissionsBoundary
- IAM:PutRolePolicy
- IAM:PutUserPermissionsBoundary
- IAM:SetDefaultPolicyVersion
- IAM:UpdateAssumeRolePolicy
- IAM:UpdateLoginProfile
- IAM:UpdateLoginProfile IAM:CreatePolicyVersion
- IAM:UpdateRole
- IAM:UpdateRoleDescription

- AWS STS:AssumeRole

To prevent privilege escalation, you should use Service Control Policies (SCPs) to prevent users in your accounts, except for IAM administrators or delegated admins, from using administrative IAM actions. If you want to safely delegate permissions management to trusted employees, you can use IAM permissions boundary feature. IAM permissions boundary allows for safe delegation of IAM permissions management while preventing escalation of privileges. For example, developers can safely create IAM roles for AWS Lambda functions and Amazon EC2 instances without exceeding certain permissions boundaries defined by the IAM administrators. Refer to the Permission boundaries round documentation for examples of permissions boundaries in action.

Monitor activity in your AWS account

Use the following guidelines to monitor your AWS account activity:

- Turn on AWS CloudTrail in each account, and use it in each supported Region.
- Store AWS CloudTrail log in a centralized logging account with very restricted access.
- Periodically examine CloudTrail log files. You can also use GuardDuty — a service that provides threat detection by continuously analyzing AWS CloudTrail Events, VPC Flow Logs and DNS Logs.
- Enable Amazon S3 bucket logging to monitor requests made to each bucket.
- If you believe there has been unauthorized use of your account, pay attention to temporary credentials that have been issued. If temporary credentials have been issued that you don't recognize, disable their permissions.

FSISEC3: How do you accommodate segregation of duties as part of your IAM role design?

Segregation of duties, as it relates to security, has two primary objectives. The first objective is the prevention of conflict of interest, abuse, and errors. The second objective is the detection of control failures that include security breaches, information theft, and circumvention of security controls.

While robust automation of infrastructure and application deployments will reduce the need for human access, there will still be instances where individuals need to complete

key functions. Segregation of duties can help mitigate risk. For users with increased privileges, it is important to distribute system administration activities so no one administrator can hide their activities or control an entire system. Additional levels of approvals for critical tasks, and independent reviews of activity are required.

Create roles by using job function policies

AWS managed job function policies can be used as a starting point to create organization-wide roles to ensure that least privilege principles are in effect. AWS provides 10 job function-based policies by default for a common set of job functions within an organization.

Use AWS Config to view historical IAM configuration and changes over time

Use AWS Config to view the IAM policy that was assigned to an IAM user, group, or role at any time in which AWS Config was recording. This information can help you determine the permissions that belonged to a user at a specific time. For example, you can view whether the user John Doe had permission to modify Amazon VPC settings on January 1, 2015.

Set up alerts for IAM configuration changes and perform audits

You can add a level of indirection by setting up alerts to notify on IAM configuration changes. This is helpful for monitoring activities by users with increased privileges. The added indirection can be set up using a combination of AWS CloudTrail, Amazon CloudWatch, and Amazon SNS. For more information, refer to the How to Receive Alerts When Your IAM Configuration Changes blog post.

FSISEC4: How do you ensure that all human access uses federation?

At financial institutions, internal and external risk and audit teams scrutinize user access management and auditability of user actions. Federation enables organizations to leverage existing functions such as user lifecycle, password, and Multi-Factor Authentication (MFA) management, to extend single sign-on for applications and the AWS Management Console.

Use federated access for developers CLI and API environments

When setting up federated access, it's important to include access to the AWS Management Console and CLI or AWS APIs. Refer to the How to Implement Federated API and CLI Access Using SAML 2.0 and AD FS blog post for details and sample

scripts for setting up SAML federation for CLI and API access. Similar scripts are also available from third-party IdP vendors.

Implement preventative and detective controls around IAM user creation

After federation is configured for human access, and EC2 instance profiles are used for application access, very few additional identities, or IAM users, are needed to be created in AWS. You may have a handful of admin identities for break glass processes (for example, if there is an issue with federation configuration or the identity provider). You may also have a handful of users for some third-party applications that do not support integration with IAM roles. Detective controls with AWS Config need to be implemented when a new IAM user or group is created.

Implement detective controls when IAM user credentials are used

Detective controls must be implemented for any API actions performed by a non-federated IAM principal. In fully federated environments that leverage IAM roles, IAM users should be used only on rare occasions, such as break glass procedures. All actions by IAM users need to be monitored and alerted on.

FSISEC5: How do you ensure that all third-party applications are accessing AWS APIs using best practices?

As a security best practice, use IAM roles and federation for third-party applications when delegating access to the organization's AWS API resources.

Grant permissions through Roles

Roles provide you temporary security credentials for the role session. A third-party application can access your AWS resources by assuming a role that you create in your AWS account. You must specify IAM permissions when defining the role's permissions policy. This policy defines the actions they can take and the resources they can access.

IAM Roles are meant to be assumed by authorized entities, such as IAM users, third-party applications, or an AWS service such as EC2. IAM Roles can be associated with EC2 instances to simplify management and deployment of AWS access keys. An EC2 instance can provide temporary security credentials to third-party applications running on that instance, which in turn can use those credentials to make API requests to your AWS resources.

Rotate and review IAM privileges assigned to third parties

Third-party applications must use federation. If federation is not supported, IAM credentials need to be rotated on a regular basis and removed after the defined purpose is no longer necessary, or if you suspect the user is compromised.

Considerations for defining the time span during which a particular IAM user needs to be rotated or removed include, but are not limited to, the sensitivity of the data, your company's security posture, corporate governance and compliance requirements, and risk of damage that a compromised IAM user could cause to your financial systems.

Infrastructure protection

FSISEC6: How do you ensure isolation between SDLC environments (dev, test, prod)?

Maintaining resource isolation between software development lifecycle (SDLC) environments reduces the chance of malfeasance and accidents in production environments. This is important guidance for all financial institutions, including those subject to Payment Card Industry Data Security Standard (PCI DSS).

Have a multi-account strategy

While you could segregate environments by deploying them in separate VPCs, deploying them in a separate AWS account provides the highest level of isolation. AWS provides patterns around multi-account strategy to handle complexity. Customers can choose to create individual accounts based on the SDLC stage, and then enforce security and infrastructure policies through this multi-account strategy. This strategy is based on the AWS Security by Design (SbD) principle — a security assurance approach that formalizes AWS account design, automates security controls, and streamlines auditing. Refer to the AWS Multi Account Strategy video for more information.

Implement IAM isolation

Having different accounts dedicated to different SDLC environments provides a natural isolation in managing privileges in IAM. AWS Organizations facilitates the management of account hierarchy. Define Service Control Policies (SCPs) to limit the actions a user can perform inside these accounts. For example, you can prevent any change in

production to CloudTrail logging, prevent internet gateways set up in a VPC, or prevent modifying AWS Config tracking.

Enforce network isolation

In addition to IAM isolation, enforce clear separation of resources between production and non-production environments. Using different accounts helps create the highest form of isolation possible on AWS. However, you may need to be able to reach resources across accounts, especially when accessing shared services such as logging and security services. VPC Peering connects resources in two VPCs (in same account or in different accounts) without the need of any additional gateway or VPN connection, and it makes all of the peered network visible to each other. This requires complete network trust between the two VPCs, and better alternatives exist depending on your use case. If the objective is to access only a few services in the other VPC, use AWS PrivateLink. AWS PrivateLink provides connectivity over an internal network without VPN and limits network exposure. Service publishers also have to specify which IAM principals can consume these endpoints and attach an IAM resources policy specifying what actions are allowed. If more extensive cross-VPC access is needed, segregation and private connectivity can be also established with AWS Transit Gateways.

FSISEC7: How do you ensure that traffic stays private whenever possible?

When deploying applications on the cloud, financial institutions can leverage Virtual Private Cloud (VPC) to carve out an isolated and private portion of the public cloud for their organizational needs. Most security-sensitive customers require that their traffic is private whenever possible and that it does not leave AWS infrastructure.

Use VPC endpoints to keep traffic private

Ensure that traffic is within the AWS infrastructure by using VPC endpoints. VPC endpoints allow private connectivity between resources in the VPC and supported AWS services. VPC endpoints allow data to be private, and they lower latency because the traffic is not routed through the internet.

Use resource policies to allow access only through the VPC endpoint

When you create an endpoint, you can attach a policy that controls the use of the endpoint to access AWS resources. For example, you restrict access to specific APIs with an endpoint policy attached to an Amazon API Gateway endpoint. Along with resource-based policies, you can also ensure that the specified AWS resources are

accessed only through the endpoint. For example, to allow access to an S3 bucket only through an endpoint, use bucket policies with a deny action to the resource if the traffic is not coming from the VPC. The `AWS:sourceVpce` condition with a VPC ID is used to specify this condition. This will force access to the bucket to be private and only go through the endpoint without traversing the internet.

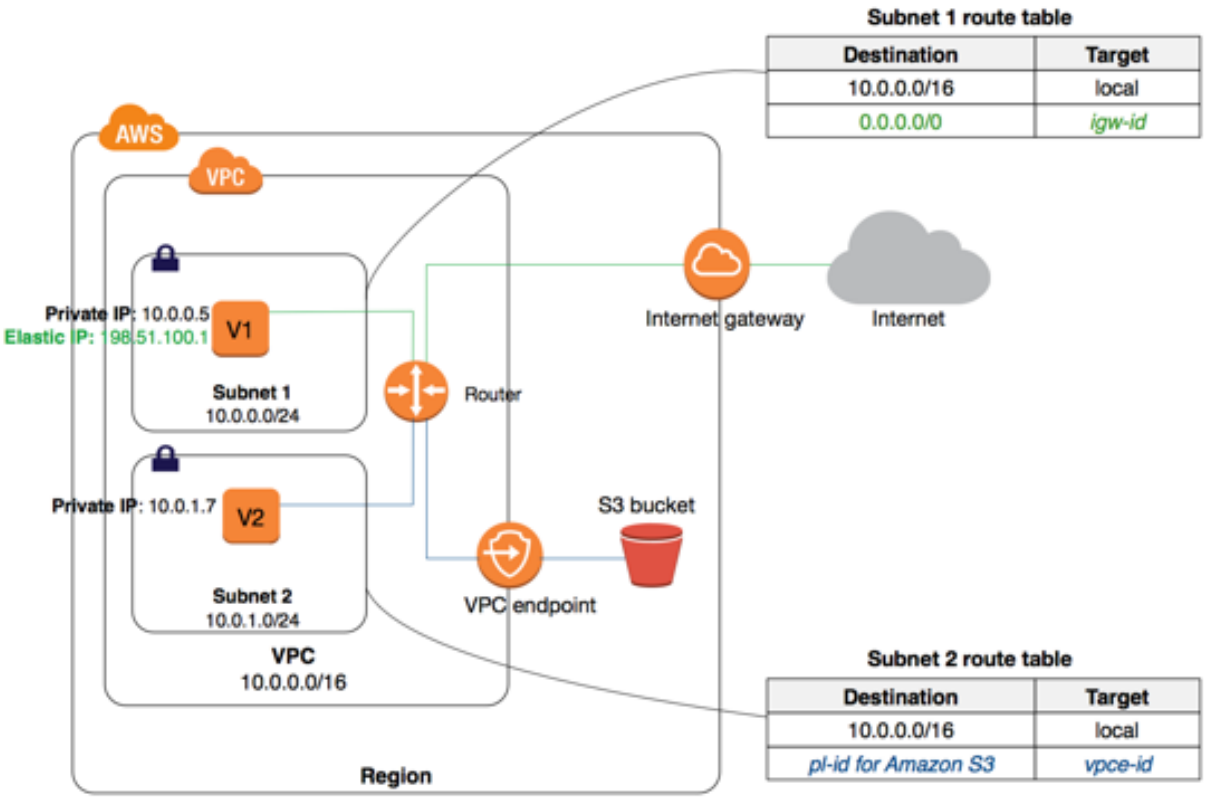

Figure 7: Instances in subnet 2 can only access Amazon S3 through the gateway endpoint

Secure databases in private subnets with restrictive security groups

Lock down databases, storage systems, and volumes in private subnets similar to what financial services customers set up behind firewalls (with no public access in their data centers). Access to databases must be limited to the application tier that is using the database on specific ports using network security groups.

FSISEC8: How are you inspecting your network for malicious traffic?

Monitor network traffic for expected and unexpected traffic to identify irregularities and gain key insight into the security of the system. For example, a poorly performing network can indicate that the network is under attack, and irregular attempts to contact unexpected external systems can indicate that an internal host has been compromised.

Monitor instance traffic

Amazon EC2 instances automatically track aggregate network inbound and outbound traffic with Amazon CloudWatch. Use custom metrics and push log files to CloudWatch for storage, aggregation, reporting, and alert notification. Create profiles for the expected network behavior for each EC2 instance and trigger alarms when deviations are detected. For example, system or web logs sent to CloudWatch Logs could trigger alarms based on the number of login failures or web request latencies. Similarly, TCP connection or outstanding connection request counts could be stored in CloudWatch and used to detect security threats like SYN flood attacks.

Monitor VPC Flow Logs for abnormal traffic patterns

Use VPC Flow Logs as a security tool to monitor the traffic that is reaching your instance, to profile your network traffic, and to look for abnormal traffic behaviors. Use VPC Flow Logs to watch for abnormal and unexpected denied outbound connection requests, which could be an indication of a misconfigured or compromised EC2 instance. CloudWatch Alerts provides rudimentary network alerting on VPC Flow Logs, and there are multiple third-party log management systems that provide extensive reporting, visualization, and alerting capabilities based on VPC Flow Log data. GuardDuty is a threat detection service that can continuously monitor your accounts by analyzing AWS CloudTrail Events, Amazon VPC Flow Logs, and DNS Logs. It also uses integrated threat intelligence such as known malicious IP addresses, anomaly detection, and machine learning to identify threats more accurately.

Flow Logs do not capture all IP traffic and have some limitations. For more information, refer to the VPC Flow Logs documentation.

Use VPC Traffic Mirroring

Use VPC Traffic Mirroring to copy network traffic from an elastic network interface of Amazon EC2 instances and forward that traffic to security and monitoring appliances for

use cases such as content inspection, threat monitoring, and troubleshooting. These security and monitoring appliances can be deployed on a fleet of instances behind a Network Load Balancer (NLB) with a User Datagram Protocol (UDP) listener. Amazon VPC traffic mirroring supports traffic filtering and packet truncation, allowing you to extract traffic that you are interested in monitoring. It also addresses challenges around having to install and run packet-forwarding agents on EC2 instances. Packets are captured at the Elastic Network Interface level, which cannot be tampered with from the user space, thus offering better security posture.

FSISEC9: How are you protecting access to your compute resources?

Use immutable infrastructure with no human access

Adopt immutable infrastructure practices with no human access to better meet your audit and compliance needs. You will be able to version control your infrastructure and handling failure will be a routine and continuous way of doing business.

Allow interactive access for emergencies only

Tightly control and monitor interactive access to EC2 instances. Interactive access should typically be provided for emergency-only, break-glass scenarios.

Test and review these pre–staged emergency user accounts, which normally are highly privileged and could be limited to read only. Limit the time duration of break-glass procedure and the password time duration. Have a ticketing system with procedure requiring that an acceptable form of authentication be provided by the requester and recorded before the accounts are made available with the aim of controlling and reducing the account's misuse, having only pre-approved personnel who will complete a certain emergency task. The break-glass accounts and distribution procedures must be documented and tested as part of implementation and carefully managed to provide timely access when needed. A special audit trail needs to be in place to monitor such emergency access for later audit and review.

You must use Systems Manager Session Manager to provide an interactive one-click browser-based shell to your Amazon EC2 instances, on-premises instances, and virtual machines (VMs). Session Manager provides secure and auditable instance management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys.

FSISEC10: How are you configuring and hardening your compute resources?

Hardening your compute resources is necessary to reduce the attack surface area of your compute resources. Ensure that the required security tools are always present, and subsequently control the deployment and lifecycle of your resources to ensure that they are always in compliance.

Build and distribute Golden AMIs

Use an automated factory to build AMIs conforming to your standards, test their compliance to required policies, probe for known vulnerabilities, and distribute them across your organization for use. Use EC2 Image Builder to create, maintain, validate, share, and deploy Linux or Windows Server images for use with Amazon EC2 and on-premises.

Deploy only what is essential

A Golden AMI will need to be hardened to run only essential software and eliminate all unnecessary processes, libraries, and tools (for example, disabling SSH access). On top of this minimal base operating system installation, you can layer additional protection software such as antivirus and endpoint protection agents, file integrity, and intrusion detection agents.

Test new AMIs for compliance with standards and known vulnerabilities using Amazon Inspector — an automated security assessment service that helps improve the security and compliance of applications deployed on AWS. Existing AMIs also need to be regularly re-tested to ensure that they are not affected by newly found vulnerabilities, as Amazon Inspector rules are regularly updated by security specialists. EC2 Image Builder also allows you to run your own tests to validate your images for functionality, compatibility, and security compliance.

Allow only approved Golden AMIs

Approved AMIs can then be distributed to your organization and tools such as AWS Organizations and AWS Service Catalog. Service Control Policies (SCPs) can be used to apply controls ensuring that new compute resources can only be started using the approved versions of the Golden AMI.

Monitor configuration changes for compliance

AWS Config rules can be used to monitor compliance to these policies, for example,

automatically highlighting older resources that are out of compliance when old AMIs are decommissioned or new vulnerabilities found.

Use your AMI pipeline for patch management

The AMI pipeline can be used to roll out patches with new versions of the Golden AMI. This strategy aligns with infrastructure as code best practices and provides a secure auditable trail for your compute resources.

FSISEC11: How are you configuring and hardening your container resources?

Protect underlying compute resources

Container security is only as good as that of the underlying host it is running on, and if that host is compromised, so are all containers running on it. For this reason, all of the advice on hardening compute resources must be followed when using containers, or use a managed service, such as AWS Fargate, which takes responsibility for the security of the container host. If you are maintaining the container hosts and associated orchestration software, use tools such as Docker Bench Security, kube-bench, and published guidelines for container security to harden your infrastructure.

Use private container repositories

Use a private container repository, such as Amazon Elastic Container Registry, to download and store your container images. A private repository allows you to maintain control over access to the repository while keeping the benefits of a centralized repository and integration with container tools. The Amazon Elastic Container Registry provides an added benefit of scanning container images against an aggregated set of Common Vulnerability and Exposures (CVEs).

Create minimal immutable container images

Start with a lightweight and secure image containing the minimal set of dependencies required to satisfy your requirements. Do not install additional software that is not needed for the operation of the container as this can introduce unexpected functionality and vulnerabilities. Container images are not meant to be modified during runtime. If a change to the image is required, perform the change through a well-implemented container build pipeline as described below.

Use a container build and deployment pipeline

Use a container build and deployment pipeline to build your container images and define stages such as tests (for example, unit and integration), code quality checks, and vulnerability scans. If your image passes all stages, tag and sign your image and upload it to the container repository triggering deployment to your environments.

Scan container images for vulnerabilities

Scan your container images as part of the CI/CD pipeline to detect and prevent vulnerabilities from being included in your deployments. Container scanning tools can detect a number of potential problems, including checking image contents against known vulnerabilities, analyzing configuration for security sensitive configurations and your own set of additional requirements. After deployment, runtime scanning of containers needs to be used to ensure continued integrity of the resources, guarding against incorrect configuration and data leaks.

FSISEC12: How do you address emerging threats?

Security-focused enterprises are taking threat identification and remediation to the next level with DevSecOps. This approach accelerates application development and ensures that threats are identified early and security testing is performed at each step of the software development lifecycle.

Automate remediation of CVEs

Scanning servers for common vulnerabilities is a long-standing best practice. However, on the cloud, customers may not only automate the evaluation of operating environments and applications, but also remediate known and emerging security vulnerabilities automatically. For example, customers may use Amazon Inspector service to automatically scan their servers in production, publish any security findings to an Amazon Simple Notification Service (SNS) topic and then create an AWS Lambda function that is triggered by those notifications to examine the findings, and implement the appropriate remediation based on the type of issue.

Perform static analysis on all code deploys

As part of a DevSecOps strategy, customers can ensure the security of their application deployments by integrating preventive and detective security controls within the pipeline. One of the key benefits of static code analysis is that you can learn about security vulnerabilities prior to provisioning AWS resources, which can help reduce costs and risk.

Conduct regular penetration testing

Simulating security incidents inside the AWS environment helps customers have a better understanding of their security posture. Financial services customers perform penetration testing of web applications most often when a new application is launched or when it's first migrated to the cloud. Some may even conduct penetration testing periodically every year. Run penetration testing regularly after every major release that involves significant re-architecture-changes. Major releases might introduce vulnerabilities that didn't exist earlier.

Deploy WebApplication firewalls

WebApplication is an application firewall for HTTP applications which applies a set of rules to an HTTP conversation. You can buy managed rule sets from the AWS Marketplace that protect against application vulnerabilities such as OWASP, bots, or emerging Common Vulnerabilities and Exposures (CVE). All Managed Rules are automatically updated by AWS Marketplace security Sellers.

Data Protection

FSISEC13: How do you classify your data?

Financial services institutions use data classification to make determinations about safeguarding sensitive or critical data with appropriate levels of protection. Regardless of whether data is processed or stored in on premises systems or the cloud, data classification is a starting point for determining the appropriate level of controls for the confidentiality, integrity, and availability of data based on risk to the organization. It is the customer's responsibility to classify their data and implement appropriate controls within their cloud environment (e.g., encryption).The security controls that AWS implements within its infrastructure and service offerings can be used by customers to meet the most sensitive data requirements.

Tag AWS services based on data classification

Data classification best practices start with clearly defined data classification tiers and requirements, which meet your organizational, legal, and compliance standards.

Use tags on AWS resources based on the data classification framework to implement compliance with your data governance program. Tagging in this context can be used for automation such as enabling and validating data encryption, retention, and archiving.

Restrict access based on classification

Use these resource tags and IAM policies, along with AWS KMS or CloudHSM, to define and implement your own policies that enforce protections based on data classification. For example, if you have S3 buckets or EC2 instances that contain or process highly critical and confidential data, tag them with a tag `DataClassification=CRITICAL` and automate data residing in them to be encrypted with AWS KMS. You can then define levels of access to those KMS encryption keys through key policies to ensure that only appropriate services have access to the sensitive content.

Leverage automated detection of confidential data

While many types of data can be classified as highly confidential, Personally Identifiable Information (PII) has long received regulatory attention. AWS offers several services and features that can facilitate an organization's implementation and automation of a data classification scheme. Amazon Macie can help you inventory and classify your sensitive and business-critical data stored in the cloud. Amazon Macie recognizes sensitive data such as personally identifiable information (PII) or intellectual property, and provides you with dashboards and alerts that give visibility into how this data is being accessed or moved.

Monitor/audit usage of data based on classification

Amazon Macie allows you to automate data protection workflows by integrating with your Security Information and Event Management (SIEM) system and Managed Security Service Provider (MSSP) solutions using CloudWatch Events. Security and compliance use cases such as alert handling, compliance ruleset creation and modifications, reporting and configurations for content in S3, and detecting user authentication and access patterns through CloudTrail can be solved with this integration. Amazon Macie also gives you the ability to easily define and customize automated remediation actions, such as resetting access control lists or triggering password reset policies.

Refer to the AWS Data Classification whitepaper for additional best practices.

FSISEC14: How are you handling data loss prevention in the cloud environment?

AWS offers a broad set of tools and services to help implement effective data protection strategies, which include IAM to prevent unauthorized access, Key Management Service (KMS), CloudHSM to manage encryption, CloudTrail to monitor data access activities, and Lambda functions to perform remediation actions in real time and Amazon Macie to monitor access patterns using machine learning.

Use Fully Qualified Domain Name (FQDN) ingress and egress filters

Specifying policies by IP may not be practical because domain names can often be translated to many different IP addresses, and maintaining security groups at each egress point can be challenging. Filtering outbound traffic by an expected list of domain names can be an efficient way to secure egress traffic from a VPC because the hostnames of these services are typically known at deployment, and the list of hosts that need to be accessed by an application are not extensive and rarely change.

Filtering traffic by a list of domain names enables companies to centralize the maintenance and deployment of rules. Use a third-party solution to implement highly available, secure FQDN Egress Filtering service.

Use VPC Endpoints and VPC Endpoint Policies for network perimeter security

VPC endpoints enable you to privately connect your VPC to supported regional services without requiring public IP addresses. When you create an endpoint, you can also attach an endpoint policy to it. This policy controls access to the service you are connecting to. VPC endpoint policies can prevent access to AWS services with non-corporate credentials by using conditions such as AWS:PrincipalAccount, AWS:PrincipalOrgId or AWS:PrincipalOrgPaths in the endpoint policy. These conditions ensure that only corporate credentials are used within the VPC to connect to your AWS regional services. Also, you can use limit access to only specific AWS resources such as specific Amazon S3 buckets through the endpoint with endpoint policies.

Enforce deny public access for S3

Use data classification best practices to identify public and non-public data. For non-public data stored in S3, make sure public access is denied. You can use the Amazon S3 Block Public Access settings on each bucket or at an account level to make sure that existing and newly created resources block bucket policies or ACLs do not allow public access. You can also define SCPs to prevent users from modifying this setting. Use AWS Config and Lambda to detect and remediate if S3 buckets are publicly accessible.

Enforce encryption

Encryption, both in transit and at rest, is another best practice to ensure the security of the data, regardless of the reason. Enabling encryption on most AWS services is simply a matter of selecting it at deployment. Use AWS Config to alert when a deployment has been made that does not meet your policy.

Configure encryption by default for S3

To avoid unintentionally storing data unencrypted, encryption for data at rest must be enabled by default. This is particularly relevant for object-based storage using S3. Set default encryption on a S3 bucket to turn on encryption by default for all objects stored in that bucket (keep in mind that any objects already stored in the bucket when encryption was turned on remain unencrypted). Use CMK-based encryption as described in FSISEC15.

Monitor VPC Flow Logs for abnormal traffic patterns

Use VPC Flow Logs to watch for abnormal and unexpected outbound connection requests, which could be an indication of unauthorized exfiltration of data. Amazon GuardDuty analyzes VPC Flow Logs, AWS CloudTrail event logs, and DNS logs to identify unexpected and potentially malicious activity within your AWS environment. For example, GuardDuty can detect compromised EC2 instances communicating with known command-and-control servers.

Audit the use of encryption in S3

In addition to setting the default encryption behavior for S3 buckets, it is important to perform periodic audits of the encryption status through automated surveillance reports. S3 Inventory reports include encryption status in its list of objects and their metadata. This is a scheduled report provided on a daily or weekly basis for a bucket or prefix. The addition of encryption status to S3 inventory allows you to see how objects are encrypted for compliance auditing or other purposes.

S3 Inventory reports can be encrypted as an extra measure of protection to prevent the objects metadata being disclosed to unauthorized parties (for example, names of files can be confidential information).

FSISEC15: How are you managing your encryption keys?

Use envelope encryption with customer managed keys

AWS KMS solution uses an envelope encryption strategy with customer master keys (CMKs). Envelope encryption is the practice of encrypting plaintext data with a data key, and then encrypting the data key under another key. Use CMKs to generate, encrypt, and decrypt the data keys that you use outside of AWS KMS to encrypt your data. CMKs are created in AWS KMS and never leave AWS KMS unencrypted.

AWS KMS supports three types of CMKs: Customer-managed CMKs, AWS managed CMKs, and AWS owned CMKs (for more information see here - https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#master_keys). For many FSI customers, Customer-managed CMK will be the preferred option because it allows for control of the permissions to use keys from any of their applications or AWS services. Customer-managed CMKs also provide added flexibility for key generation and storage. In addition, every use of a key or modification to its policy is logged to AWS CloudTrail for auditing purposes.

Rotate encryption keys

Cryptographic best practices discourage extensive reuse of encryption keys. Security best practice is to enable automatic key rotation for an existing CMK. When you enable automatic key rotation for a customer-managed CMK, AWS KMS generates new cryptographic material for the CMK every year. AWS KMS also saves the CMK's older cryptographic material so it can be used to decrypt data that it encrypted.

Automatic key rotation has no effect on the data already encrypted with a CMK. It will neither change existing CMK data keys nor re-encrypt any data protected by that the CMK, and it will not mitigate the effect of a compromised data key. In this case, data will have to be re-encrypted with the new data key.

Monitor encryption logs

Monitoring the logs of encryption key usage and administration activities is a critical feature in the financial services industry. AWS KMS also works with AWS CloudTrail to provide encryption key usage logs to help meet your auditing, regulatory, and compliance needs.

Monitor key deletes

Key destruction can only be performed by the key administrators. Ensure that all destruction requests are reviewed within the safety window – (a key cannot be

destroyed immediately. It is disabled which prevents use and is deleted at the expiry of the window).

FSISEC16: How are you securing credentials for your applications (for example, database connection string and login info)?

A recent Netwrix survey called 'IT Risks in Finance: Danger of Human Errors' identified that 78% of data loss incidents are caused by regular users. Insecure passwords and password sharing were identified as the top two security threats.

Secure credentials for your applications

Application credentials (database credentials, passwords, tokens, and API Keys) grant access directly or indirectly to customer data. For example, you will have to occasionally store credentials and passwords somewhere in order to authenticate your application with the database. Protecting application credentials with appropriate mechanisms helps reduce the risk of accidental or malicious use and unauthorized access to sensitive financial data that must be kept secure and in compliance with your company's security policies.

Store application credentials securely

Managing application secrets like database credentials, passwords, or API Keys are easy when you're working locally with one machine and one application. As you grow and scale to many distributed microservices, it becomes a daunting task to securely store, distribute, rotate, and consume secrets. Previously, customers needed to provision and maintain additional infrastructure solely for secrets management, which could incur costs and introduce unneeded complexity into systems.

Use AWS Secrets Manager to automatically update and rotate your credentials. Your secrets are encrypted with the AWS Key Management Service (KMS) key of your choice, and administrators can explicitly grant access to these secrets with granular IAM policies for individual roles or users.

Alert when secrets are in code (plaintext or encrypted)

Do not hardcode credentials on the client side of applications. Implement mechanisms ensure confidence that if the code became open source, your credentials would not be compromised.

A common practice is to encrypt the secrets in your code, thus not exposing their values in your source control, or to other developers. However, to decrypt those secrets, the server must manage another key. This “secret decryption” key must be stored and accessed securely. We recommend transitioning away from this approach to storing secrets externally in cryptographically secure Secrets Manager.

Store configuration data securely at scale

Use AWS Systems Manager to provide a centralized store to manage your configuration data. Parameter Store allows you to separate your secrets and configuration data from your code. Your configuration can be encrypted with the KMS key of your choice, and you can explicitly grant access to these parameters with granular IAM policies.

Run application processes under the principle of least privileges

Applications running under elevated privileges (superuser) on your OS can be used by attackers to run toolsets capable of mining your application server for data or uncovering application flaws. Even legitimate users with permissions to run processes with superuser privileges could compromise your application using hidden key loggers or other malware. When deploying and running your applications always define granular permissions to control which resources it should have access to. For example, the application should only have access to files and folders that it needs. Enable operating systems capabilities such as kernel security modules for Linux (Secure Enhanced Linux, AppArmor, etc.) if available. Similarly, containers should not run as privileged containers unless required for specific purposes such as security monitoring/scanning. Use features such as Linux Capabilities in Amazon ECS or pod security policies in Amazon EKS to enforce the principle of least privilege at runtime.

Separate privileged and non-privileged processes

There are cases where an application requires elevated privileges to perform specific operations. In these scenarios, you must separate the privileged from the unprivileged and run them as separate processes. Each process or service can then be configured to run under different security principals in accordance with the principle of least privilege.

FSISEC17: How do you audit access and use of secrets?

You must fully document access controls by role for sensitive financial services applications and data related to the financial reporting chain.

Monitor and audit your credentials to ensure that the usage of your secrets and any changes to them are logged. This ensures that any unexpected usage or change can be investigated, and unwanted changes can be rolled back. Use AWS Secrets Manager to store your secrets. AWS Secrets Manager currently supports other AWS services (CloudWatch and CloudTrail) that monitor your organization's secrets and the activity that happens within it.

Audit secret access

AWS Secrets Manager integrates with AWS CloudTrail. Using the information collected by CloudTrail helps you determine when each request was made to Secrets Manager, the IP address from which the request was made, who made the request, and additional details.

Monitor secret use

Secrets Manager works with CloudWatch Events to trigger alerts when administrative operations occur on secrets. For example, you could warn administrators when a secret is deleted or when a secret is rotated. Configure CloudWatch Events rules to look for these operations and send the generated events to administrator defined targets — an Amazon SNS topic or a simple AWS Lambda function that's triggered by the event, which logs the details of the operation for your later review.

Monitor secret versions scheduled for deletion

Use a combination of AWS CloudTrail, Amazon CloudWatch Logs, and Amazon Simple Notification Service (Amazon SNS) to create an alarm that notifies you of any attempts to access a version of a secret that is pending deletion. If you receive a notification from such an alarm, cancel deletion of the secret to give yourself more time to determine whether you really need to delete it. Your investigation might result in the secret being restored because it really is still needed. Alternatively, you might need to update the user with details of the new secret they should use instead.

FSISEC18: How are you protecting the integrity and security of your logs?

Auditability of logs and the assurance that logs cannot be tampered with is important for financial services to demonstrate the effectiveness and compliance of their operational controls. Audit trails or logs are important in identifying incidents of non-compliance. In this section, we describe services and best practices that help ensure the ongoing integrity of logs. These best practices may also address regulatory requirements for retention, indexing, and accessibility of data, including logs.

Enable log file integrity validation for CloudTrail

You can determine whether a log file was modified, deleted, or unchanged after CloudTrail delivered it, by enabling log file integrity validation for CloudTrail. CloudTrail creates a separate digest file every hour that contains hashes for each of the files delivered in the last hour and signs the digest file with a public/private key pair.

Use AWS Config

Enable AWS Config to track changes to resources configuration, answer questions about resource configurations, demonstrate compliance either at a specific point in time or over a period of time, troubleshoot, or perform security analysis. When processing configuration change notifications, leverage AWS Lambda or Amazon Simple Queue Service (SQS) with workers to process, filter, and consolidate change notifications and alerting.

Use Amazon S3 Object Lock

You can securely deliver logs to a designated S3 bucket, and use the S3 Object Lock feature to ensure immutability of logs. S3 Object Lock is an S3 feature that blocks object version deletion during a customer-defined retention period so that you can enforce retention policies as an added layer of data protection. In conjunction with S3 Versioning, which protects objects from being overwritten, you're able to ensure that objects remain immutable for as long as S3 Object Lock protection is applied. You can apply S3 Object Lock protection by either assigning a "Retain Until" date or a "Legal Hold" to an object. You can apply retention settings within a PUT request, or apply them to an existing object after it has been created.

FSISEC19: How are you protecting against ransomware?

Ransomware is a form of malware that encrypts a victim's files. The attacker then demands a ransom from the victim to restore access to the data upon payment. The

malware can come in through different routes such as email attachments, downloads from websites or as a part of a larger more sophisticated data breach. However, once foothold is established, the malware can start encrypting any data in sight. There are two parts to an effective strategy against ransomware: prevention and recovery.

Prevent malware infiltration by securing compute resources

Protect compute resources to make sure that they are patched against vulnerabilities that can be exploited to install malware. A number of best practices discussed under FSISEC9, FSISEC10, FSISEC11 and FSISEC12 to ensure that your compute resources can be patched with automation and monitored for compliance.

Prevent attacker from accessing your data stores

Scoping access to data based on the principal of minimum privileges will help prevent as well as limit the blast radius of an attack. An effective data classification scheme, along with enforcement and monitoring based on that scheme as discussed in FSISEC14 can help prevent an attacker from having accessing and encrypting your data.

Network isolation and segregation is another effective protection as compromised systems cannot reach deep into your network. FSISEC7 discusses a number of best practices to ensure your data stores are accessed over a private network, from a limited number of hosts.

Detect attacks with monitoring data access failures

You need to integrate data access logs (for example S3 Access Logs) into your SIEM Monitor and alert on repeated access failures – these could be the canaries for an attack. Unusual CPU utilization spikes can also indicate that an attacker re-encrypting your data – data encryption is CPU intensive.

Use frequent backups to recover from an attack

Because ransomware makes itself known quite quickly, incorporate short-lived anti-ransomware backups into your backup cycle. AWS makes it easy to take snapshots of data stores, so back up often and keep these around for only a few days to limit costs. FSISEC19 discusses strategies that can be used to protect the integrity of your backups.

Key AWS Services

- **Identity and Access Management**
  - **AWS Directory Service**: Integrate Active Directory-dependent workloads, such as Amazon EC2 for Microsoft Windows Server or Amazon RDS for SQL Server, custom .NET applications, and AWS Enterprise with Microsoft Active Directory.
  - **AWS Identity and Access Management (IAM):** Control users' access to and usage of AWS. Create and manage users and groups and grant or deny access. Enforce strong authorization and authentication.
  - **AWS Organizations**: Centrally manage the creation and policies applied to multiple AWS accounts.
- **Detective Control**
  - **AWS CloudTrail:** Enable governance, compliance, operational auditing, and risk auditing of your AWS account. Log, continuously monitor, and retain events related to API calls across your AWS infrastructure.
  - **AWS Config:** Facilitate resource inventory, configuration history, and configuration change notifications to enable security and governance.
  - **Amazon CloudWatch:** Monitor AWS Cloud resources and applications running on AWS, collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes to your AWS resources.
- **Data Protection**
  - **AWS CloudHSM (Cloud Hardware Security Model):** Meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated Hardware Security Module appliances.
  - **AWS Key Management Service (KMS):** Create and control the encryption keys used to encrypt your data.
- **Infrastructure Security**
  - **Amazon EC2 Systems Manager:** Helps you automatically manage inventory, apply OS patches, create secure system images, and configure secure operating systems.
  - **AWS Certificate Manager:** Provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates.

  - **AWS Shield:** Thwart DDoS attacks by configuring select AWS services to build a solution or employ our DDoS-dedicated managed service.
  - **AWS Web Application Firewall (AWS WAF):** Protect your web applications from common web exploits that could impact availability, security, and resources.
  - **Amazon Inspector:** Employ automated security assessments that help improve the security and compliance of applications deployed on AWS.
  - **Amazon Virtual Private Cloud (VPC):** Provision a logically isolated section of AWS where you can launch AWS resources in a virtual network that you define.

- **Incident Response**
  - **AWS Config Rules:** Allows you to create rules that automatically act in response to changes in your environment, such as isolating resources, enriching events with additional data, or restoring configuration to a known-good state.
  - **GuardDuty:** Provides threat detection that continuously monitors malicious activity and unauthorized behavior in your AWS accounts and workloads. GuardDuty analyzes tens of billions of events across multiple AWS data sources, such as CloudTrail Events, VPC Flow Logs and DNS Logs and makes it easy to automate how you respond to threats leveraging Amazon CloudWatch Events and AWS Lambda.
  - **Amazon Detective**: Simplifies the process of investigating security findings and identifying the root cause. Amazon Detective analyzes trillions of events from multiple data sources such as VPC Flow Logs, AWS CloudTrail logs, and Amazon GuardDuty findings and automatically creates a graph model that provides you with a unified, interactive view of your resources, users, and the interactions between them over time.
  - **AWS Lambda:** Use our serverless compute service to scale your programmed, automated response to incidents.

Reliability Pillar

The **reliability pillar** includes the ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues.

The technology systems of financial institutions are complex and highly interconnected to each other and to non-financial entities. Payment processing, trading and settlement, market data, custody and entitlement management, and financial messaging are examples of the types of applications on which the proper functioning of many industries depend. Regulators continue to focus on the resiliency of financial institutions through bodies such as the Basel Committee on Banking Supervision, Board of Governors of the Federal Reserve System, and Bank of England.

In this section, we will provide in-depth best practices that financial institutions can use with AWS services to construct highly elastic, highly available, resilient, and scalable solutions at lower costs compared to traditional on-premises IT. To discuss these best practices, we use the concept of service availability interchangeably with the Recovery time objective (RTO) and Recovery Point Objective (RPO). An introduction to the concept of service availability and its relation to the recovery objectives can be found in the Well-Architected Reliability Pillar.

Financial institutions can leverage AWS services to provide the levels of resiliency and availability that their applications need to provide. The AWS Global infrastructure is built around Regions and Availability Zones (AZs). AWS services differ in their scope and availability – while some services are only available in a Single-AZ (Amazon EC2, Amazon EBS), others span across multiple Availability Zones in a Region (S3) and some offer Cross-Region Replication capabilities for even greater levels of availability. Some AWS services, such as CloudFront and Route 53, are deployed in our Edge network outside of Regions. A more comprehensive background of availability and scope of AWS infrastructure can be found in the Resilient Applications for Financial Services whitepaper.

Design for Resiliency

AWS offers capabilities that can be leveraged to provide different levels of resiliency in the cloud. The implementation, configuration, and operation of applications on AWS is the customer's responsibility. Financial institutions must use the following dimensions when building resilient applications on the cloud:

- Resiliency requirement planning
- Resiliency architecture
- Monitoring
- Development and deployment

- Data backup and retention

Resiliency Requirement Planning

FSIREL1: How do you determine the resiliency requirements for your workload?

Use business criticality to drive Recovery Objectives

A key to determining resiliency requirements is to establish the criticality of any function a workload supports. Financial institutions must also consider any regulatory requirements around resiliency and architect applications with those requirements in mind. Financial institutions place greatest scrutiny on critical functions, which can include services provided by financial services institutions to external end users or participants where a disruption to the service could cause intolerable harm to consumers or market participants, harm market integrity, threaten policyholder protection, safety and soundness, or financial stability.

The resiliency requirements put around important business services needs to be proportionate to their importance. This needs to be reflected in the risk appetites set by the financial institution, which inform recovery targets (RTO, RPO) and availability metrics. Financial institutions need to architect applications such that their resiliency keeps risks within their stated appetite, and monitor availability so that it remains so on an ongoing basis. They also need to test the reliability of their controls and recovery capabilities to bring a risk back within appetite in the event it materializes and that they are able to continuously operate despite disruption.

Apply fine grained application resiliency requirements

It's common to initially think of an application's availability as a single target for the application as a whole. However, upon closer inspection, we frequently find that certain aspects of an application or service have different availability requirements. For example, some systems might prioritize the ability to receive and store new data ahead of retrieving existing data. Other systems prioritize real-time operations over operations that change a system's configuration or environment. The Well-Architected Reliability Pillar whitepaper outlines a few of the ways that you can decompose a single application into constituent parts and evaluate the availability requirements for each. The benefit of decomposing is to focus efforts on availability according to specific needs, rather than engineering the whole system to the strictest requirement.

Cost is an important factor, and designing high levels availability can be very expensive. Separating the most critical parts from others can allow you to make effective cost tradeoffs as well as providing the capability to have degraded performance while still providing critical functionality.

Use past examples of market volatility in determining peak loads

In financial services workloads, even ones that do not directly provide services for traders such as settlement and clearing, market volatility creates peak demand requirements with a “long-tail” — the peak volume of an extreme event is much higher than one would expect to model a normal distribution and thus typical p95 and p99 metrics are insufficient for estimating peak load. Determine if the workloads have dependencies on market volatility and adjust load testing scenarios based on historical peaks. It is common that financial services workloads are subject to dramatic increases in demand. The scaling response to the increase in demand must keep up with the change in demand. For example, automatic scaling can take several minutes to add capacity, and workloads with rapid changes in demand may exceed the ability of automatic scaling to react. Resiliency requirements need to be created, remembering that failures can occur during periods of excess loads.

Resiliency architecture

FSIREL2: Is your architecture designed for resiliency?

Understanding how AWS services impact your workloads availability is in an important step in determining the resiliency of your architecture.

Build resilient Tier 1 applications using best practices

Workloads designated by regulators and financial institutions as Tier 1 are subject to greater scrutiny because they must demonstrate that they can recover within a short downtime with very little data loss. To achieve those targets, you must leverage automated operations, consistent deployments, and predictive monitoring with proactive responses in addition to Resilient Application Design Patterns. These strategies are described in the Resilient Applications on AWS for Financial Services whitepaper.

FSIREL3: Does your network support your resiliency requirements?

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a private, isolated section of the AWS Cloud where you can launch AWS resources in a virtual network. However, there are a number of features that can influence how resilient workloads react to network failures, loss of connectivity, unexpected increase in network traffic, or DOS attacks. Networking design underpins the capacity of any workload to meet its resiliency targets. The Well-Architected Reliability pillar whitepaper outlines foundational best practices in the design of networking.

Establish baselines for expected network traffic

To understand conditions of high or unexpected network traffic, you must establish a baseline of metrics for the expected data flows between users and systems. This baseline should trigger an operational response when a workload is under a DOS attack or under unexpected load. AWS provides many services that can provide protection against DOS attacks. AWS Shield and AWS Shield Advanced provide an integrated web application firewall (AWS WAF) for web applications running on Amazon CloudFront, ELB and EC2 resources. AWS virtual networking features like VPC security groups and network access control lists (ACLs) are also effective in protecting against network attacks.

Monitoring

FSIREL4: How do you monitor your resources?

High availability for applications requires the ability to detect failures and recover quickly. Applications must be configured to emit the relevant telemetry to detect failures, so that operational processes can capture and react to these events.

Use a single pane of glass for monitoring

While AWS Cloud services provides robust monitoring, you must organize the data to escalate issues as quickly as possible. Without adequate processes in place, you may miss leading indicators of problems. A single pane of glass and standardizing cloud monitoring standards across your organization, can help avoid information silos and simplify the analysis of monitoring data. Combining monitoring of AWS, system metrics, and application logs enables analysts to cross-reference signals and log information across dependent systems. Frequently, issues surface in invoking systems, and IT

professionals spend time parsing logs on the invoking systems instead of on the dependent systems where the error originated.

Alert on the absence of an event

The absence of monitoring data can indicate an underlying issue. Implement controls that alert on missed reporting intervals. Treat missing data as breaching and raise alarms.

Identify metrics and validate alerts through load testing

Workloads must be load-tested regularly to validate scaling and resiliency. Identify key metrics (for both components that auto scale with demand and/or resources that do not such as relational databases) that correlate with capacity constraints and customer outage during these load tests. Create alerts and dashboards to monitor.

Include validation of automated alerts and remediation as part of application testing. Run load tests in lower environments, identify triggers for alerting and verify the effectiveness of the automated remediation. If a workload's Mean Time to Detection (MTTD) can be minimized, there is more time for the recovery mechanism to respond and availability of your applications will increase.

Review runbooks and RCA to identify new alarms and automate remediation

At regular intervals, review your operational runbooks and incident playbooks to identify commonly occurring manual processes. Create alerts that trigger these everyday activities and implement automation that executes the steps in the playbook. It is essential to perform detailed root cause analysis on these issues to provide the most effective remediation.

Use distributed tracing tools for service-oriented architectures

As systems become more interdependent with the implementation of microservices architectures, the challenge of identifying performance bottlenecks increase. Use application performance monitoring tools such as AWS X-Ray to trace and provide telemetry across multiple systems. AWS X-Ray is an integrated tool that supports serverless, containers, and on-premises workloads providing tracing and execution data as transactions span across multiple services.

AWS Backup and Retention

FSIREL5: How are you backing up data in the cloud?
FSIREL6: How are backups retained?

Understand requirements for data backup and retention

An important task of determining the resiliency requirements of a workload is to identify data backup and retention needs. Financial institutions may have standards for backup and retention of data in their systems, which may be informed by regulatory requirements. Financial services customers must understand the requirements that apply to the workloads that are running in their environments.

Back up logs as part of the backup strategy

In addition to the backup of application data and databases, the system logs may also fall under regulatory requirements. Include the CloudTrail, CloudWatch Logs, applications, and system logs in the log backup plan. In AWS, customers use S3, Amazon S3 Glacier, EBS snapshots, and RDS snapshots for backups of AWS services, and AWS Storage Gateway for on-premises backup to AWS. The AWS Backup service centralizes the management of the backups across the AWS environment by creating tag-based policies to manage the backups.

Incorporate anti-ransomware backups into your backup strategy

In addition to the normal backup cycle, short-lived anti-ransomware backups need to be inserted into the backup cycle. These extra backups only need to be held for a day or two since ransomware makes itself known quite quickly. This limits the extra storage costs. Define a backup cycle and retention period for protection against ransomware attacks. A regional copy of the data is sufficient but access to these backups must be highly restricted (backups are also encrypted as the sources should be). Refer to FSISEC19 for a more detailed discussion around preventing ransomware attacks

Create lifecycle policies for backups

Based on regulatory requirements, create lifecycle policies to retain and purge data in AWS. For data in S3, S3 Lifecycle policies allow of the automation of migration of data to the most appropriate storage tier. AWS Backup allows for the management of retention of data across the environment through tag-based policies.

Use S3 Object Lock for WORM storage

Financial institutions can use S3 Object Lock mode to store data using a "write-once-read-many" (WORM) model. The Amazon S3 Object Lock mode applied to an object prevents any user from modifying that object. To track which objects have S3 Object Lock, you can refer to an S3 Inventory report that includes the status of objects. Amazon S3 Object Lock helps you meet regulatory requirements that require WORM storage, or simply add another layer of protection against object changes and deletion. Amazon S3 Object Lock has been assessed by Cohasset Associates for use in environments that are subject to SEC 17a-4, CFTC, and FINRA regulations. For more information about how Amazon S3 Object Lock relates to these regulations, see the Cohasset Associates Compliance Assessment for Amazon S3 whitepaper.

Key AWS Services

- **Resilient Architecture**
  - **Amazon S3:** Leverage Amazon S3 object storage to provide durability and resiliency of your data on AWS. It is available Regionally (resilient against events that impact an entire Availability Zone) and also supports Cross-Regional replication for geographic isolation.
  - **EC2 Auto Scaling:** Maintain application availability and automatically add or remove EC2 instances according to conditions you define. You can also use the dynamic and predictive scaling features of EC2 Auto Scaling to respond to changing demand as well schedule the right number of EC2 instances based on predicted demand to scale faster.
  - **Amazon Route 53:** Use the availability of Route 53 to direct traffic based on latency, proximity and application health checks to enable a variety of low-latency, fault-tolerant architectures.
  - **AWS Direct Connect:** Connect your data centers to AWS over dedicated, private and consistent connection using DX.

- **Amazon Virtual Private Cloud (VPC)**: Provision a logically isolated section of AWS where you can launch AWS resources.
- **Amazon CloudFront:** You can cache your content in CloudFront's edge locations worldwide and reduce the workload on your origin by only fetching content from your origin when needed. You can use CloudFront's native origin failover capability to automatically serve your content from a backup origin when your primary origin is unavailable.
- **Amazon RDS Multi-AZ:** Use RDS Multi-AZ deployments to provide enhance availability for production database workloads. RDS synchronously replicates primary instance to a secondary in a different AZ which runs on a physically distinct independent infrastructure. In case of infrastructure failure, RDS automatically fails over to the standby so that you can resume database operations.
- **Amazon DynamoDB:** Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB automatically spreads the data and traffic for your tables over a sufficient number of servers to handle your throughput and storage requirements and all of your data is stored is automatically replicated across multiple Availability Zones in an AWS Region.
- **AWS Shield and AWS Shield Advanced:** AWS Shield is a managed service that provides protection against Distributed Denial of Service (DDoS) attacks for applications running on AWS**.** AWS Shield Advanced provides additional protections against more sophisticated and larger attacks for your applications running on Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, and Route 53.
- **AWS Lambda:** AWS Lambda lets you run code without provisioning or managing servers. AWS Lambda is designed to use replication and redundancy to provide high availability for both the service itself and for the Lambda functions it operates. There are no maintenance windows or scheduled downtimes for either.

- **Monitoring**
  - **CloudWatch:** Amazon CloudWatch is the principal monitoring service for AWS Cloud resources and the applications you run on AWS.

  - **VPC Flow Logs:** VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. VPC Flow Logs can be monitored through CloudWatch.

- **AWS Backup and Retention**
  - **Amazon S3 Glacier**: Amazon Simple Storage Service Glacier, is an extremely low-cost storage service optimized for infrequently used data, or "cold data".
  - **EBS snapshots, and RDS snapshots**: Snapshots for both RDS and EBS allow point in time recovery of the data stored in them. They can be configured to run automatically or at a scheduled time.
  - **AWS Backup:** Is a centralized backup service that makes it easy and cost-effective for you to back up your application data across AWS services in the AWS Cloud and on premises. Storage volumes, databases, and file systems are backed up to a central place where you can configure and audit the AWS resources you are backing up, automate backup scheduling, set retention policies, and monitor all recent backup and restore activity.

Performance Efficiency Pillar

The **performance efficiency** pillar focuses on the efficient use of computing resources to meet requirements and on maintaining that efficiency as demand changes and technologies evolve.

Regulators expect financial services institutions to define operational performance objectives for workloads, and implement policies that help achieve those objectives. The objectives must define both qualitative and quantitative measures of operational performance and explicitly state the performance standards that the workload intends to meet. In this section, we focus on strategies for the efficient use of computing resources to meet those requirements, and how to maintain that efficiency as demand changes and technologies evolve.

After objectives and service-level targets are defined, you need to regularly monitor and assess whether or not the workload meets expectations. The workload's performance must be reported regularly. Operational objectives are also expected to be reviewed regularly and incorporate new technology and business developments.

FSIPERF1: How do you select the best performing architecture?

Performance objectives for workloads can vary depending on the criticality of the workload. While more stringent performance requirements are expected for critical systems such as market data feeds, trade execution, settlement, and clearing systems all cloud workloads can benefit from defining performance requirements.

Use internal and external risk to determine performance requirements

External regulatory, as well as internal risk requirements, are often a good place to start for performance requirements. For some systems, regulators release sector-wide guidance including any potential stress tests; but for others, regulators require that financial institutions have the capability to deliver on the operational resilience and the performance targets they have set for themselves.

Factor in rate of increase in load and scale-out intervals

Identify the upper bounds of the peak load against a system, as well as the amount of time needed to reach peak load. Load tests often overlook the rate of increase in traffic and create tests that scale up too quickly or too slowly. If the load test ramps up too quickly, the system may not be able to add capacity rapidly enough to meet the demand, which will degrade performance and introduce errors. Load tests need to be run periodically and with every major release of the system.

FSIPREF2: How do you evaluate compliance requirements?

Monitoring ensures that you are aware of any deviation from expected performance.

Use Application Performance Monitoring (APM)

Using an APM provides your organization the capability to ensure application performance meets its defined requirements. AWS offers features to monitor and right-size the cloud services that you need to meet performance requirements. For example, you can monitor and set alarms on latency and error rates for each user request, or on all your downstream dependencies, or on the success and failure of key operations. This level of monitoring generates huge amounts of data which can be challenging for operation teams to store, analyze, and visualize. Teams frequently need training to update their skills and processes and take full advantage of this new fidelity of insight.

Verify consistency and failure recovery during load tests

You must verify data consistency and recovery during periods of high load. Ensuring that your workload's RTO and RPO will still be valid under the highest load can uncover

gaps in your architecture and operational resilience.

Include dependencies in your load tests

Financial institutions need to map resources they need to continuously deliver their important business services. These resources are your people, processes, technology, facilities, and information, including third-party service providers. This mapping allows the identification of operational dependencies, vulnerabilities, and threats. Incorporating the dependencies of your workload (such as on Financial Messaging providers) as part of your performance tests will enable you to demonstrate the overall resiliency of your workload.

Cost Optimization Pillar

The **cost optimization** pillar focuses on how to architect systems with the most effective use of services and resources at a minimal cost. Cost optimization can be challenging in traditional on-premises solutions because you must predict future capacity and business needs while navigating complex procurement processes. Adopting the practices in this pillar ensures that you will be able to build architectures that can:

- Ensure that your usage and costs move in line with demand.
- Use appropriate services and resource types to minimize costs.
- Analyze, attribute, and forecast costs.
- Reduce costs over time.

Cost optimization is a continual process of refinement and improvement of a system over its entire lifecycle. A cost-optimized system will fully utilize all resources, achieve an outcome at the lowest possible price point, and meet your functional requirements.

Proactive vs. reactive cost optimization

As with other pillars within the Well-Architected Framework, there are trade-offs. For example, whether to optimize for speed-to-market, or for cost. Sometimes, it’s necessary to optimize for speed in order to go to market quickly or to meet a deadline. We often see customers overcompensate “just in case” rather than spend time benchmarking for the most cost-optimal deployment. This can lead to over-provisioned and underutilized resources. However, this may be a necessary choice when you need to “lift and shift” from your on-premises environment to the cloud, and then optimize afterwards.

No matter which option you choose, financial services customers must invest in a cost optimization strategy in order to realize the economic benefits of the cloud more readily. The Well-Architected Cost Optimization Pillar whitepaper describes techniques and best practices for both the initial and the ongoing cost optimization deployment in your environments.

Conclusion

The goal of the Financial Services Industry Lens for the Well-Architected Framework is to provide architectural best practices for designing and operating reliable, secure, efficient, and cost-effective regulated financial services workloads on AWS. In operational excellence, we outline best practices around how people, process and operating models need to be aligned so that workloads running on AWS can support critical financial services business services. Architectures for financial services workloads need to incorporate security and evidence-based compliance design patterns. Financial services customers also need to continuously monitor, measure, test failure and recovery in the cloud to achieve their business resiliency and performance objectives. These objectives can be met with significant cost savings by right sizing and establishing governance models around consumption and monitoring of AWS resources.

This Framework can improve security, resiliency, and operational efficiency for all financial services customers migrating and building apps on AWS, and can also assist in meeting regulatory and compliance obligations.

Contributors

Contributors to this document include:

- Arjun Chakraborty, Principal Solution Architect, AWS Financial Services
- Ilya Epshteyn, Principal Solutions Architect, AWS Financial Services
- Misha Goussev, Principal Solutions Architect, AWS Financial Services
- Som Chatterjee, Senior Technical Program Manager, AWS Commerce Platform
- James Craig, Senior Partner Solution Architect, AWS Financial Services
- Anjana Kandalam, Manager, Solutions Architecture, AWS
- Roberto Silva, Senior Solutions Architect, AWS

- Chris Redmond, Senior Consultant, Governance, Risk and Compliance, AWS Professional Services
- Pawan Agnihotri, Senior Manager, Solutions Architecture, AWS Global Financials
- Rahul Prabhakar, Global FSI Lead, AWS Security Assurance
- Jaswinder Hayre, Senior Manager, Solutions Architecture – Security, AWS
- Jennifer Code, Principal Technical Program Manager, AWS Financial Services
- Igor Kleyman, FSI Industry Specialist, AWS Security Assurance
- John McDonald, Head of Governance, Risk & Compliance – Americas, AWS Financial Services
- John Kain, Head of Banking and Capital Markets Business Development

Document Revisions

Date	Description
June 2020	Updated question numbering in FSISEC and FSIREL. Minor text updates to improve accuracy.
May 2020	First publication