



AWS  
re:Invent

**SEC338**

# How FINRA builds serverless data-masking pipelines across accounts

**Geetha Ramachandran**

Director, Application Engineering  
FINRA

**Latha Nagaraj**

Principal Application Architect  
FINRA

# Agenda

- 01 | FINRA introduction
- 02 | Setting the context
- 03 | Challenges
- 04 | Solution and demo
- 05 | Metrics and impact
- 06 | Future roadmap and Open source
- 07 | Q&A

# Key session takeaways

- Learn about how **FINRA securely enables application teams** to get a copy of production data with automation and pipelines
- Learn about **Serverless AWS services** used in the automation solution
- Learn about **why data obfuscation** is necessary
- Learn about data obfuscation **considerations and best practices**
- Learn about various other **security controls** put in place to protect sensitive data and adhere to compliance

# FINRA mission



Investor  
protection



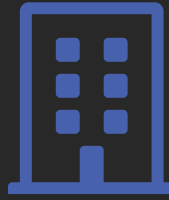
Market  
integrity

# Volume



12

Markets/exchanges



3,800

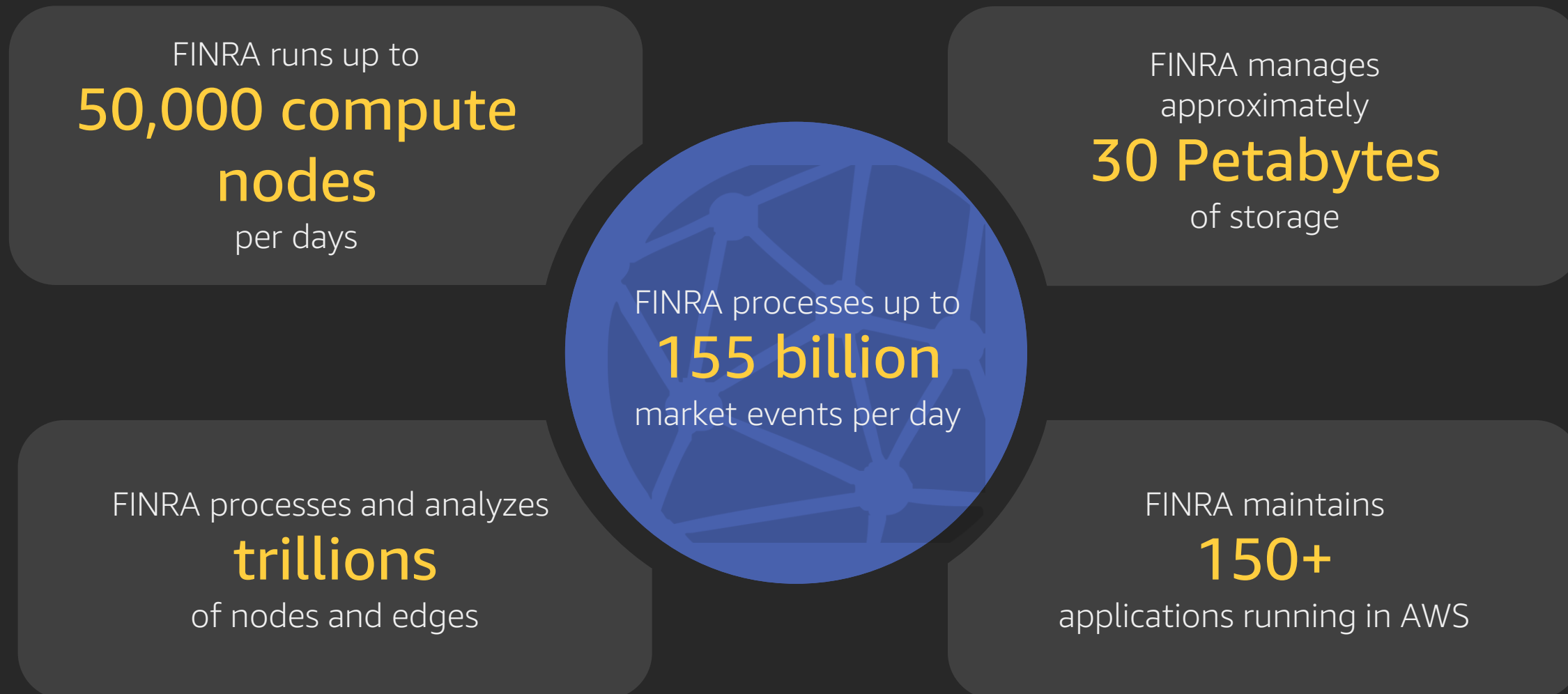
Firms



634,000

Brokers

# Big data



# Setting the context



# Use cases

## Development

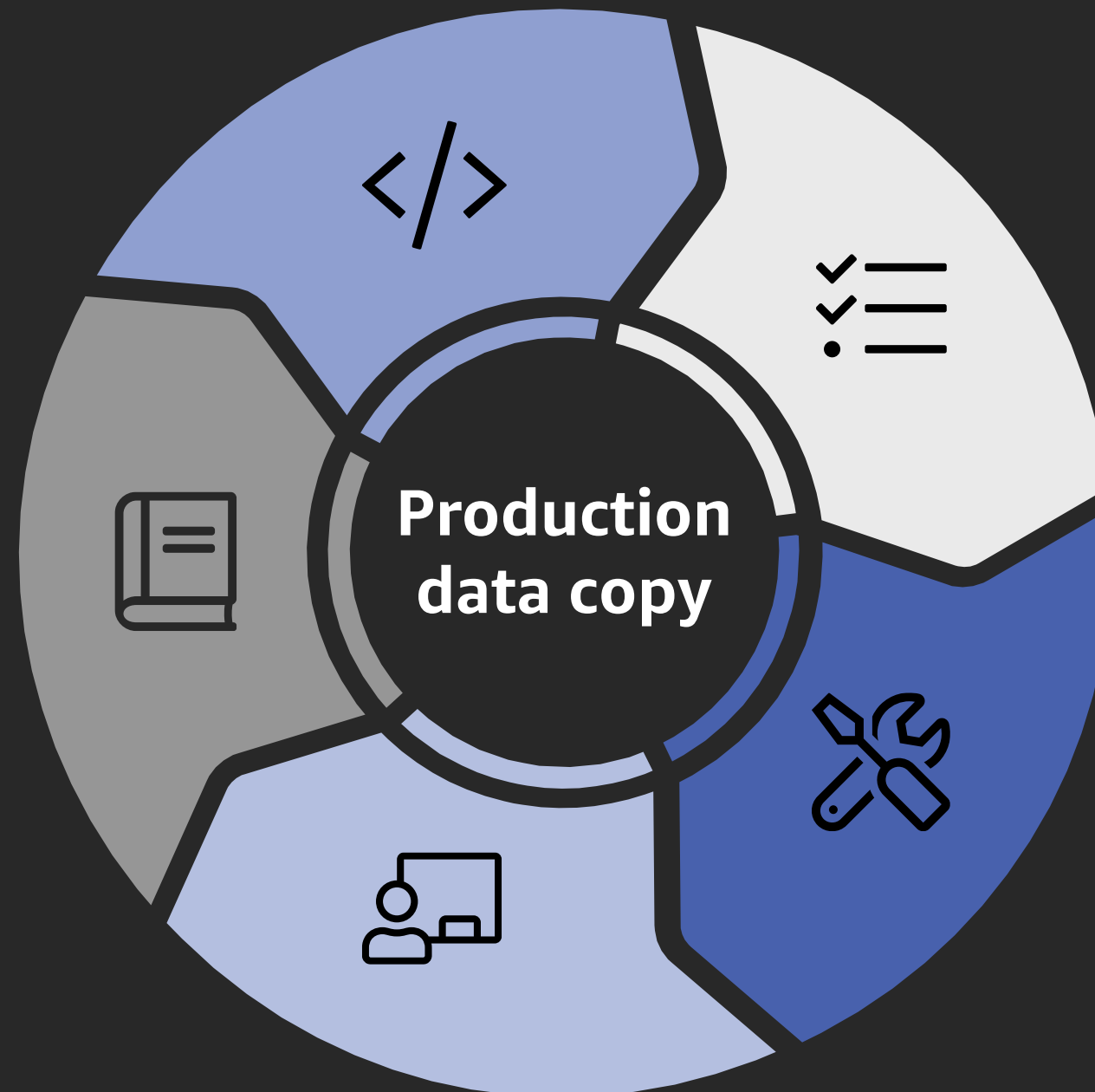
- Design
- Implementation

## Research

- Identify gaps
- Normalize data

## Training

- External users
- Internal users



## Testing

- Functional
- Reliability
- Performance
- User acceptance

## Troubleshooting

- Early detection
- Defects

# Challenges in production data copy

---

## Security



Sensitivity of data



Enforcing compliance



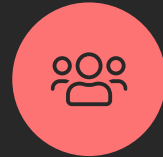
Traceability/  
auditability

---

## Operational



Data volume



Multiple AWS accounts

---

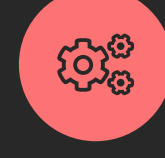
## Productivity



Frequency of masking



Turnaround time



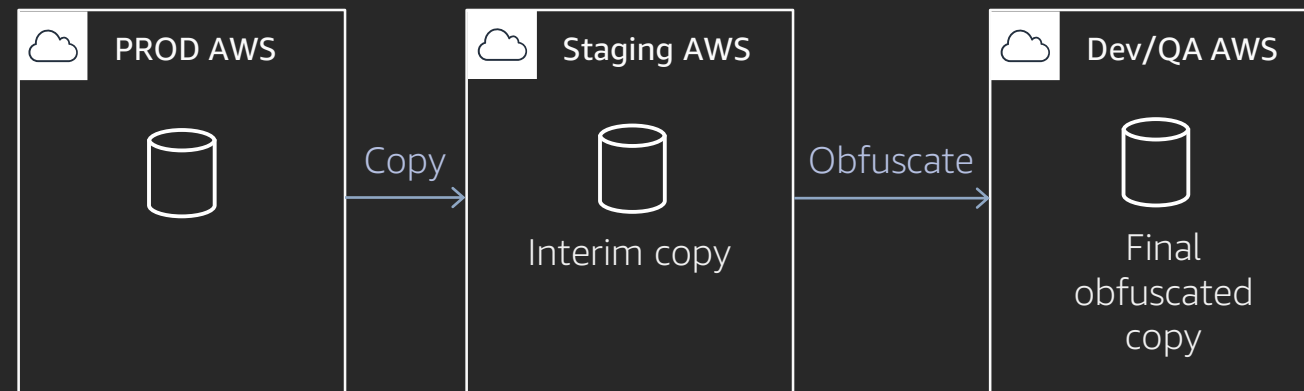
Lack of automation

# Challenge: Handling sensitive data

Obfuscate sensitive data while retaining structure and format

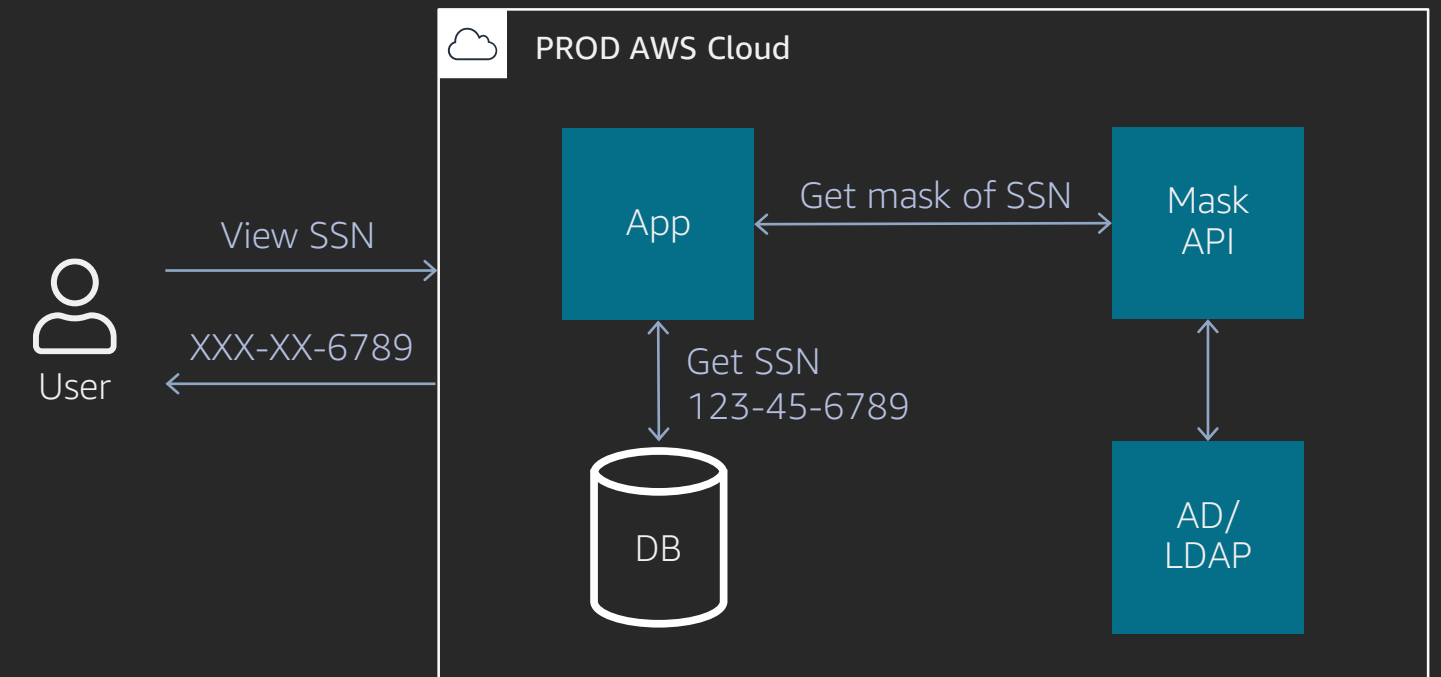
## Data obfuscation

Copy data and obfuscate the copy



## Real-time data masking

Data masked in transit



# Solution

# Design considerations



## Self-service model

Copy anytime



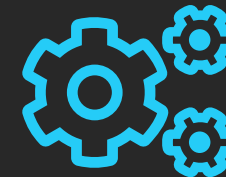
## Security controls

Data classification risk checked  
AppSec reviewed



## Data obfuscation

Obfuscate at-rest  
Plug points for custom obfuscation  
tools



## E2E automation

One-click pipelines



## Serverless framework

Lightweight

# Solution: Maskopy

- ✓ Data obfuscation framework
- ✓ Obfuscates snapshots from production and copies to lower environments
- ✓ Built on AWS serverless services



AWS Step Functions (orchestration service to coordinate microservices using visual workflows)

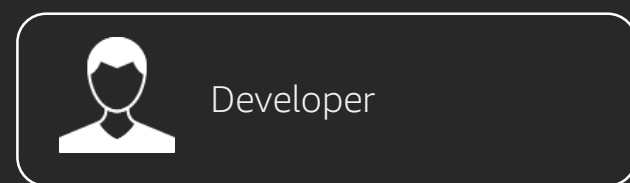


AWS Lambda (compute service that lets you run code without provisioning or managing servers)

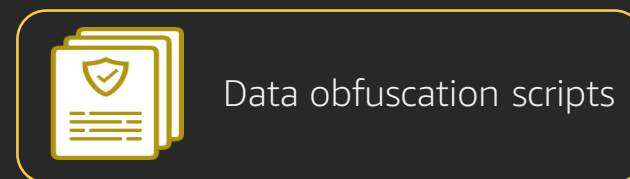


AWS Fargate (compute engine to run containers without managing servers or clusters)

# Maskopy workflow

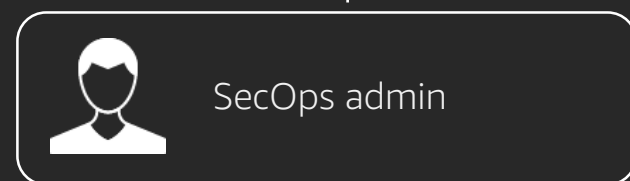


Provide

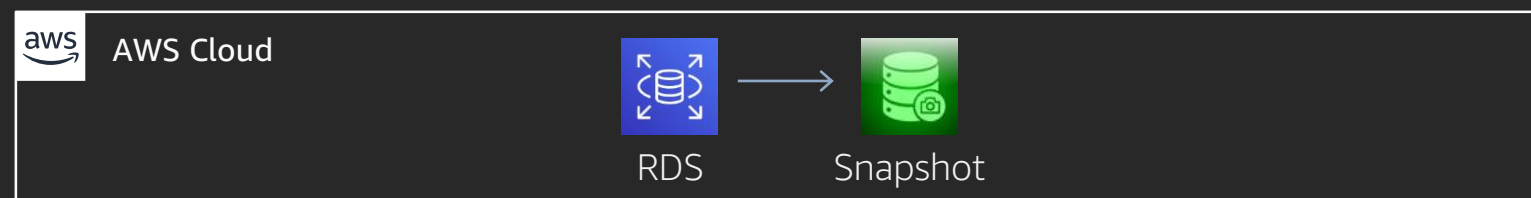


Submit

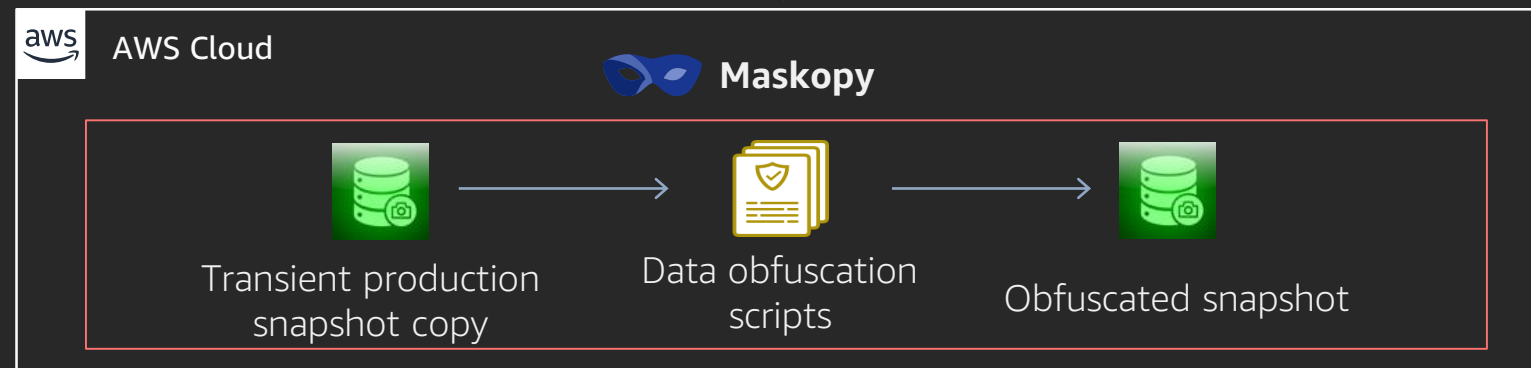
Review



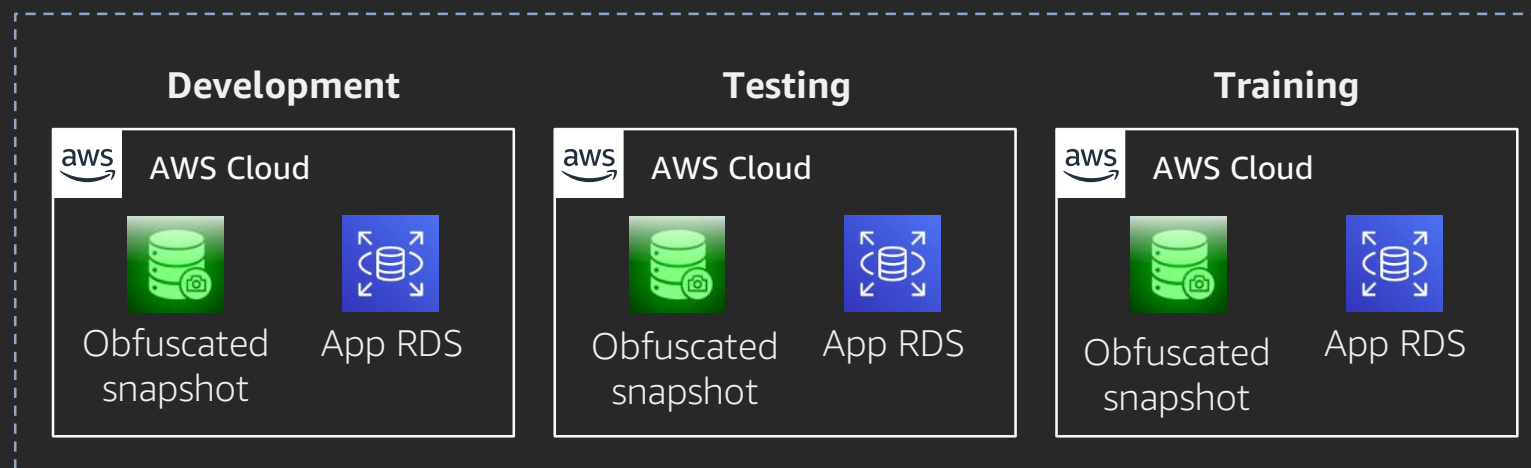
## Production account



## Data-staging environment



## Target accounts



# Data obfuscation script

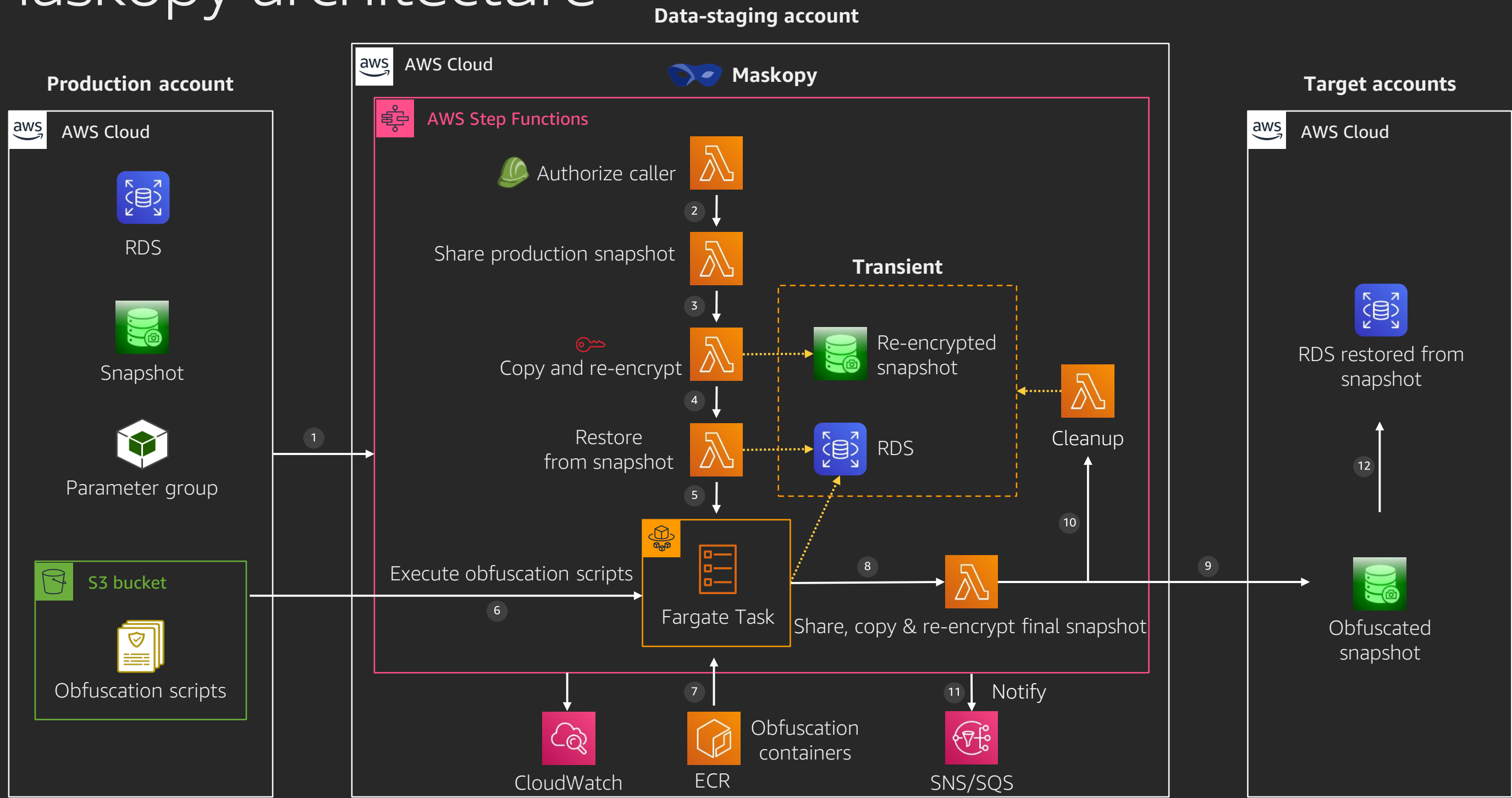
```
obfuscatedb --encrypt  
            --use-local-config  
            --db ${DB_NAME}  
            --table EMPLOYEE  
            --column BIRTH_DT  
            --id test@test.com --format AUTO
```

- ✓ Data obfuscated at rest
- ✓ Reduced risk
- ✓ Lower volume of sensitive data



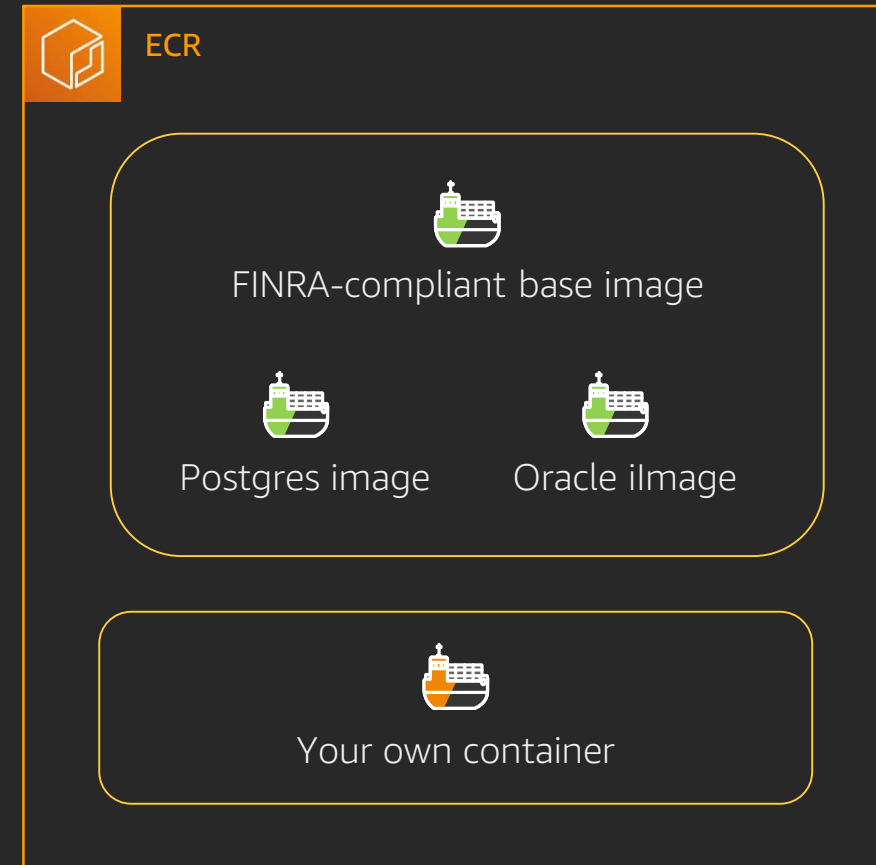
# Architecture

# Maskopy architecture



# Obfuscation containers

- ✓ Reusable, compliant FINRA base image
- ✓ Vendor specific images with required tools and reusable obfuscation scripts
- ✓ Bring your own container (BYOC)  
(different obfuscation tools, database vendors)



# Data obfuscation considerations



## Irreversible

- ✓ Cannot change back to original data



## De-identification

- ✓ Protect confidentiality of individuals
- ✓ Minimize risk of unintended disclosure of identity



## Referential integrity

- ✓ Consistently obfuscated data
- ✓ Consistent even across several heterogeneous data sources



## Algorithms

- ✓ Industry-standard
- ✓ NIST approved or recommended
- ✓ Stay updated with emerging methods



## FPE

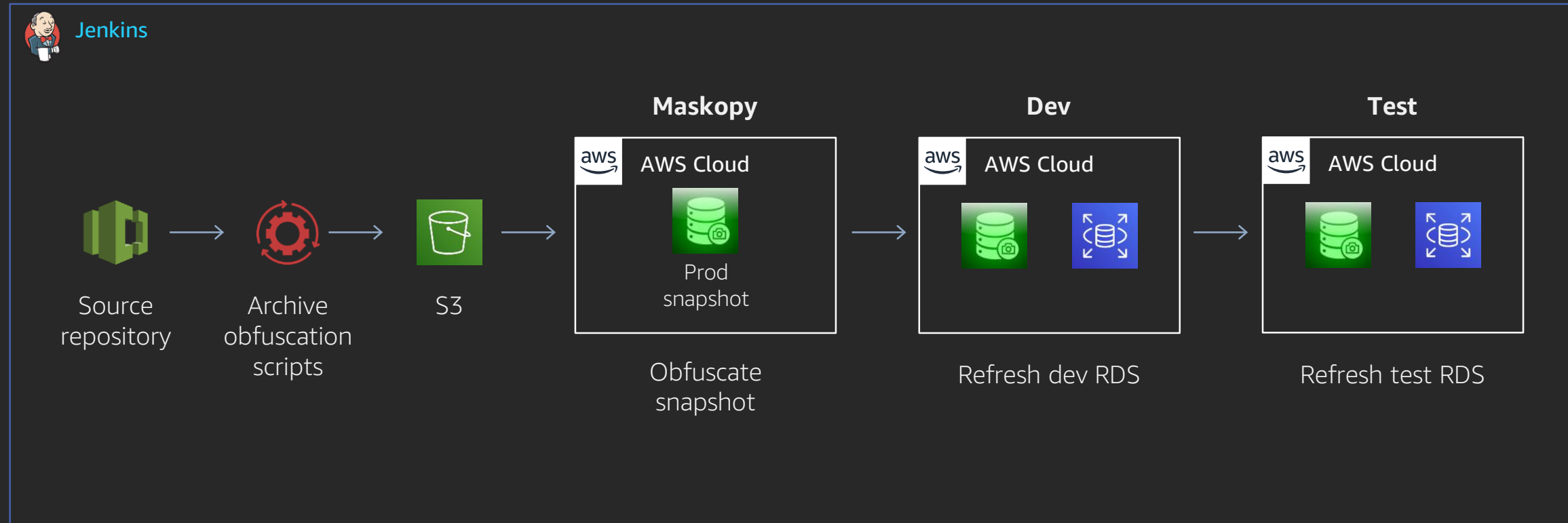
- ✓ Format-preserving encryption
- ✓ Retain structure and format



## Realistic values

- ✓ Fictional but realistic
- ✓ Retain realism of production data

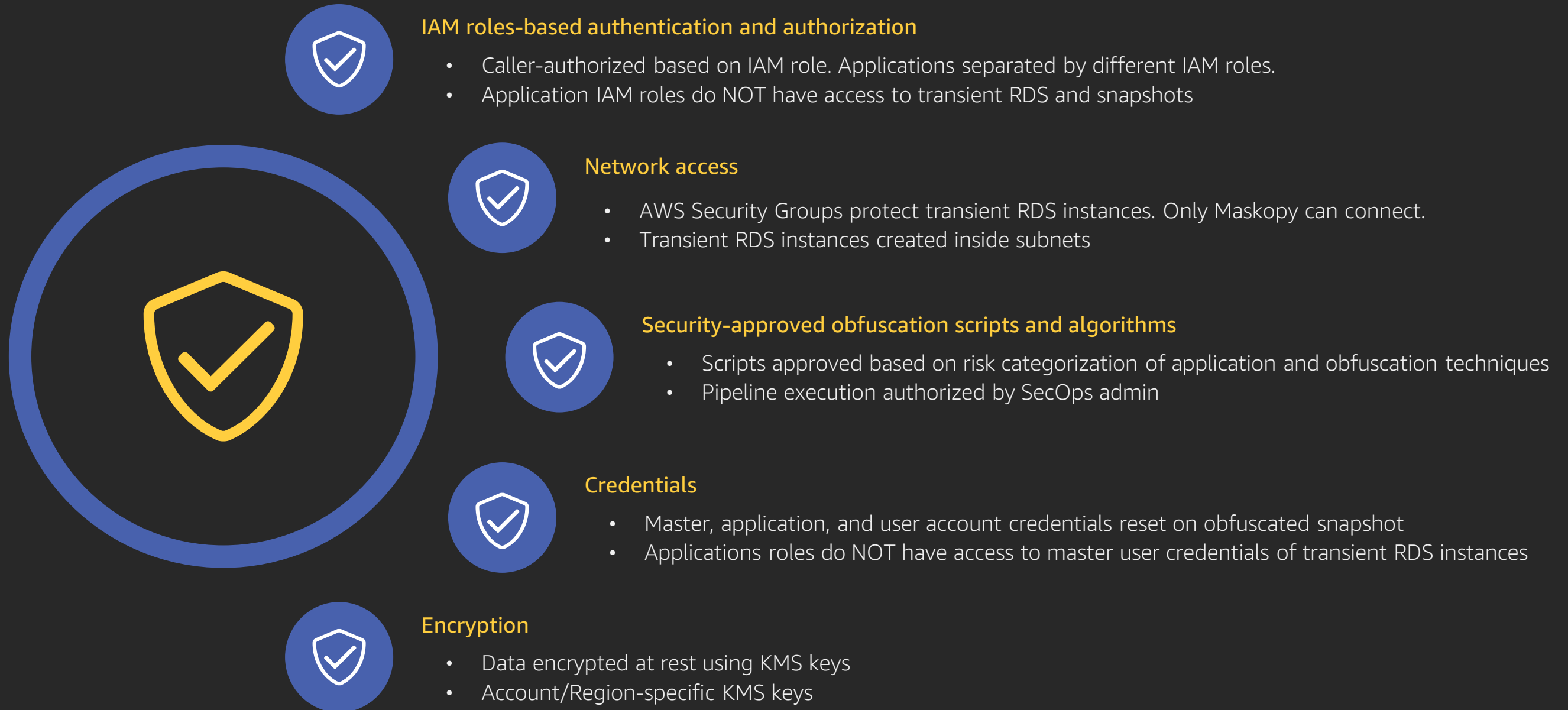
# Automation pipeline



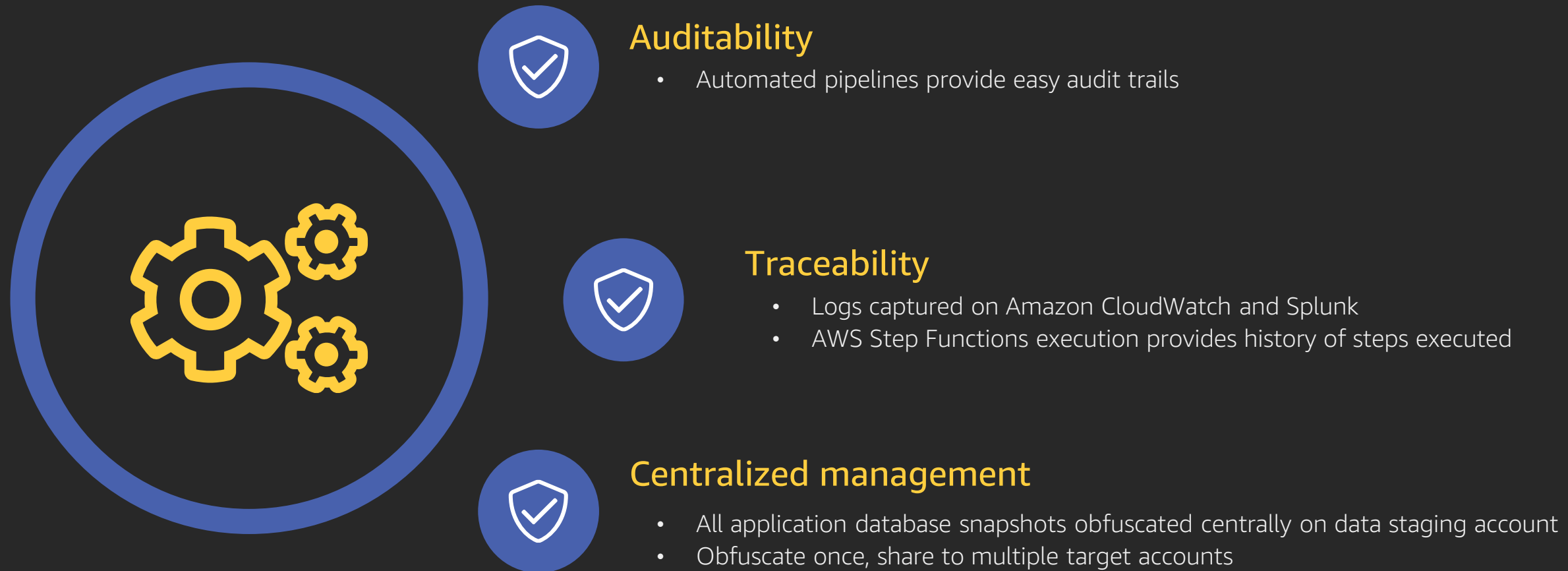
- ✓ Security-approved data obfuscation scripts staged once
- ✓ Lower environment databases refreshed periodically with obfuscated production data

# Demo

# Five security controls

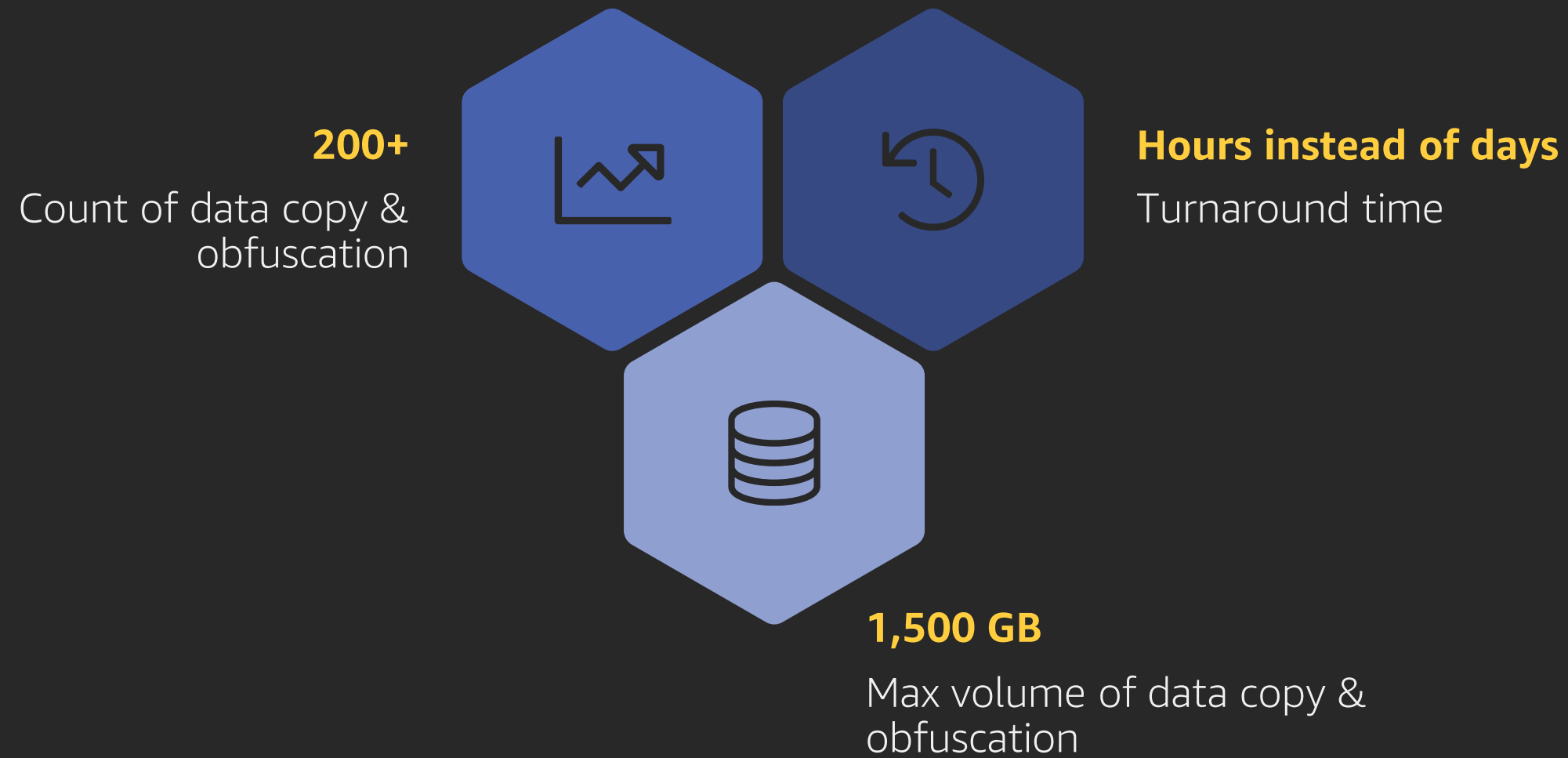


# Operations and governance





# Metrics

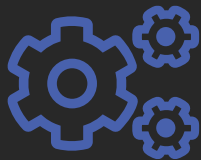


# Impact



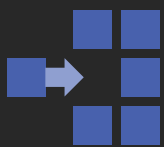
## Cost efficient

- Short-lived transient RDS/ECS instances for obfuscation
- No long-running servers



## Operations

- Lightweight serverless solution and orchestration
- Centralized service, low maintenance in terms of upgrades and patching



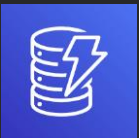
## Pluggability

- Easy integration with other automation/solutions
- Bring your own obfuscation container

# Future



Support Amazon Aurora RDS




Support Amazon DynamoDB



Observability on security controls

# Open source

 Maskopy

Overview

Maskopy

Overview

Quick Start

AWS Setup

IAM Role Setup

Configurations

Quick Start

This quick start will help us get started with build, deployment and execution of Maskopy Step Function. The Step Function executes a series of lambda functions and runs a fargate task to obfuscate data.

Pre-requisites

This section lists the tools for local setup and AWS resources that are needed for Maskopy to run.

Local Environment

Need below tools in the local to build lambda functions, connect to AWS accounts and build docker container for obfuscation.

1. python3, pip, zip

2. aws cli

3. docker

AWS Environment

Maskopy requires a minimum of two AWS accounts.

1. Source Account : This account hosts the RDS instance within a vpc and subnet which needs to be masked and copied to other accounts. Take a note of vpc-id and subnet-ids. These will be required to provide as inputs when creating resources in this AWS account. **IAM Roles** : Follow the steps documented in [Setup in Source account](#) and create IAM roles in source account.

2. Staging Account : This account is where the lambdas and step function are deployed and executed from. The final snapshot gets created in this account. This account needs to have a vpc, public and private

Contents

Pre-requisites

Local Environment

AWS Environment

Create Resources in AWS

Execution

Step 1 : Export AWS credentials

Step 2 : Provide Input

Step 3 : Run Maskopy

# FINRA open-source projects

<http://technology.finra.org/opensource.html>

---

## Available



### Maskopy

Serverless data  
obfuscation solution



### Gatekeeper

Temporary access  
to EC2/RDS



### yum-nginx-api

GO API to upload  
RPM to Yum



### Aphelion

Monitor  
AWS service limits



### Fidelius

Secrets  
management in AWS

---

## Coming soon



### CloudPass

Temporary token for  
AWS



### Portus

Security group  
manager



### Provision

Create resources in  
AWS

# Learn security with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate cloud security skills



30+ free digital courses cover topics related to cloud security, including Introduction to Amazon GuardDuty and Deep Dive on Container Security



Classroom offerings, like AWS Security Engineering on AWS, feature AWS expert instructors and hands-on activities



Validate expertise with the **AWS Certified Security - Specialty** exam

Visit [aws.amazon.com/training/paths-specialty/](https://aws.amazon.com/training/paths-specialty/)

# Thank you!

**Geetha Ramachandran**

Geethalaksmi.Ramachandran@finra.org

**Latha Nagarajsai**

Latha.Nagarajsai@finra.org



Please complete the session  
survey in the mobile app.