AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

COM204

Building data lakes in highly regulated markets

Martijn van Dongen
Cloud Evangelist, AWS Community Hero
Schuberg Philis

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Disclaimer

I’m an AWS generalist, not a legal expert, data analyst, data steward, or data engineer, and I’m not from AWS

The best practices shown in this presentation are not reviewed or approved by any regulatory – yet

In this presentation, I use shortcuts and omit details to avoid showing sensitive implementation details

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Data analytics refresher

Use case

Architecture

11 best practices

Next steps

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Refresher

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS data lakes refresher

Catalog & search



AWS Glue



Amazon DynamoDB

Data ingestion



Amazon Kinesis Data Firehose



AWS Database Migration Service (AWS DMS)

Central storage



Amazon Simple Storage Service (Amazon S3)

User access



AWS AppSync



Amazon API Gateway



Amazon Cognito

Processing & analytics



Amazon RDS



AWS Lake Formation



Amazon QuickSight



Amazon EMR



Amazon Redshift



AWS Glue



Amazon Athena

Security & governance



AWS Key Management Service (AWS KMS)



AWS CloudTrail



AWS Identity and Access Management (IAM)



Amazon CloudWatch

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Regulators

DNB (De Nederlandsche Bank/Dutch National Bank)

EBA (European Banking Authority)

PCI DSS (Payment Card Industry Data Security Standard)

GDPR (General Data Protection Regulation)

EEA (European Economic Area)

AFM (Authority Financial Markets)

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Compliance dominators

Risk analysis

Exit strategy

Data processing agreement (DPA)

Patterns and anti-patterns

The right to be forgotten

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Are AWS services compliant?

	PCI	SOC	eu-central-1
Amazon Athena	✓	✓	✓
Amazon Redshift	✓	✓	✓
AWS Glue	✓	✓	✓
Amazon S3	✓	✓	✓

Source: https://aws.amazon.com/compliance/services-in-scope/

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case: Fund your next house (FYNH)

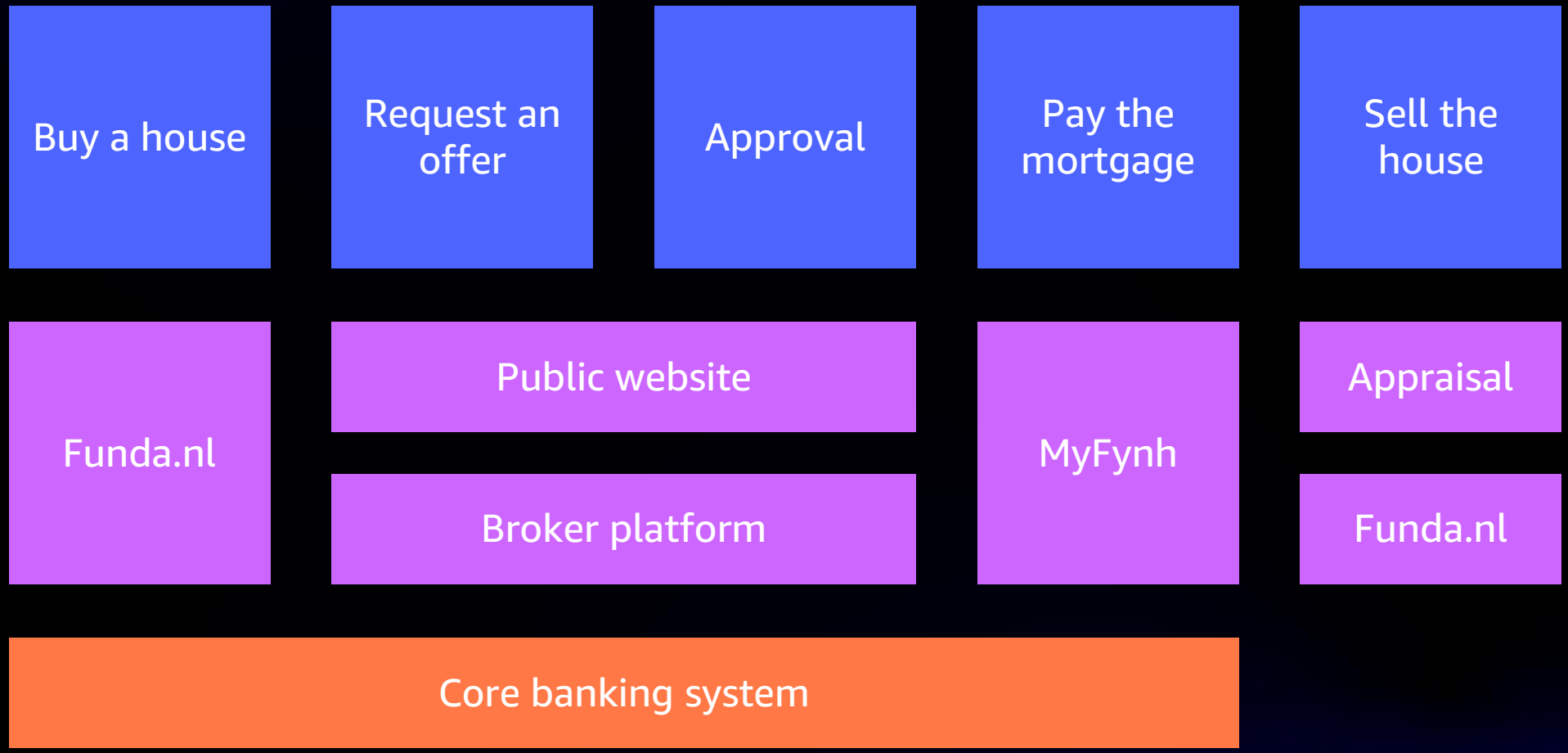
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case: Customer journey



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Use case systems



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Design principles

Cloud native first

Serverless first/low ops

Highest security standards

Least privilege

4-eyes principle

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

High-level data lake architecture

Funda.nl

Broker platform

Public website

MyFynh

Ingestion



Amazon Kinesis Data Firehose



AWS Database Migration Service (AWS DMS)

Data lake



AWS Glue



Amazon Simple Storage Service (Amazon S3)

Analytics



Amazon QuickSight



Amazon Athena

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11 best practices to build a data lake in highly regulated markets

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11 best practices

Security

1. Use 4 buckets
2. Encrypt all data
3. Data science studio
4. Using a query pipeline

Access control

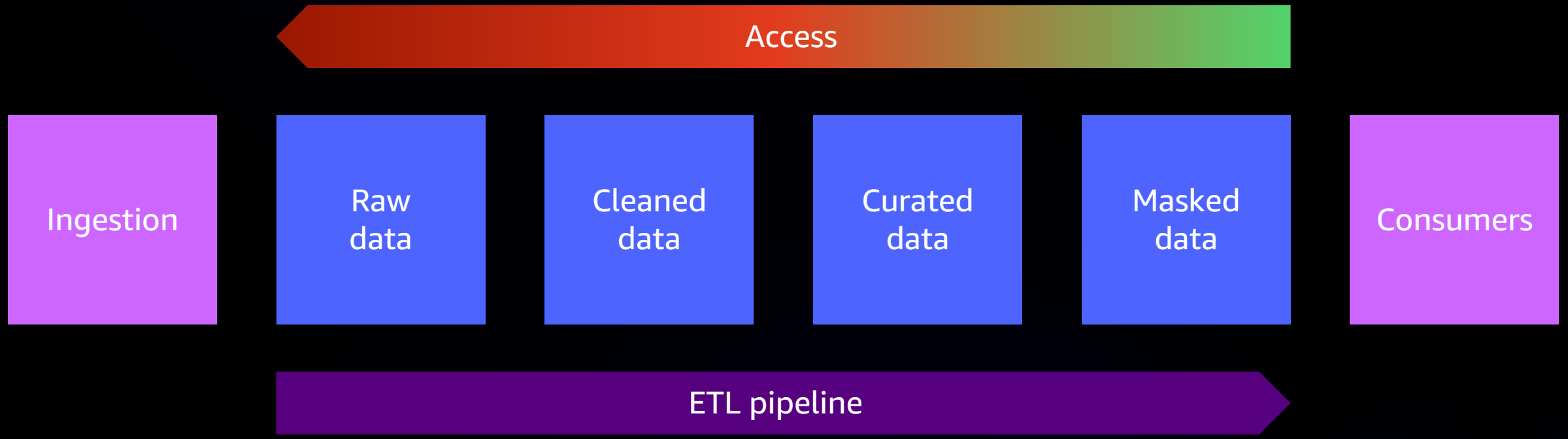
5. Infrastructure as code
6. Break-glass procedure
7. Working with metadata
8. Fine-grained access control

Compliance

9. Deleting data in the data lake
10. Detect violations with Amazon Macie
11. Data compliance dashboard

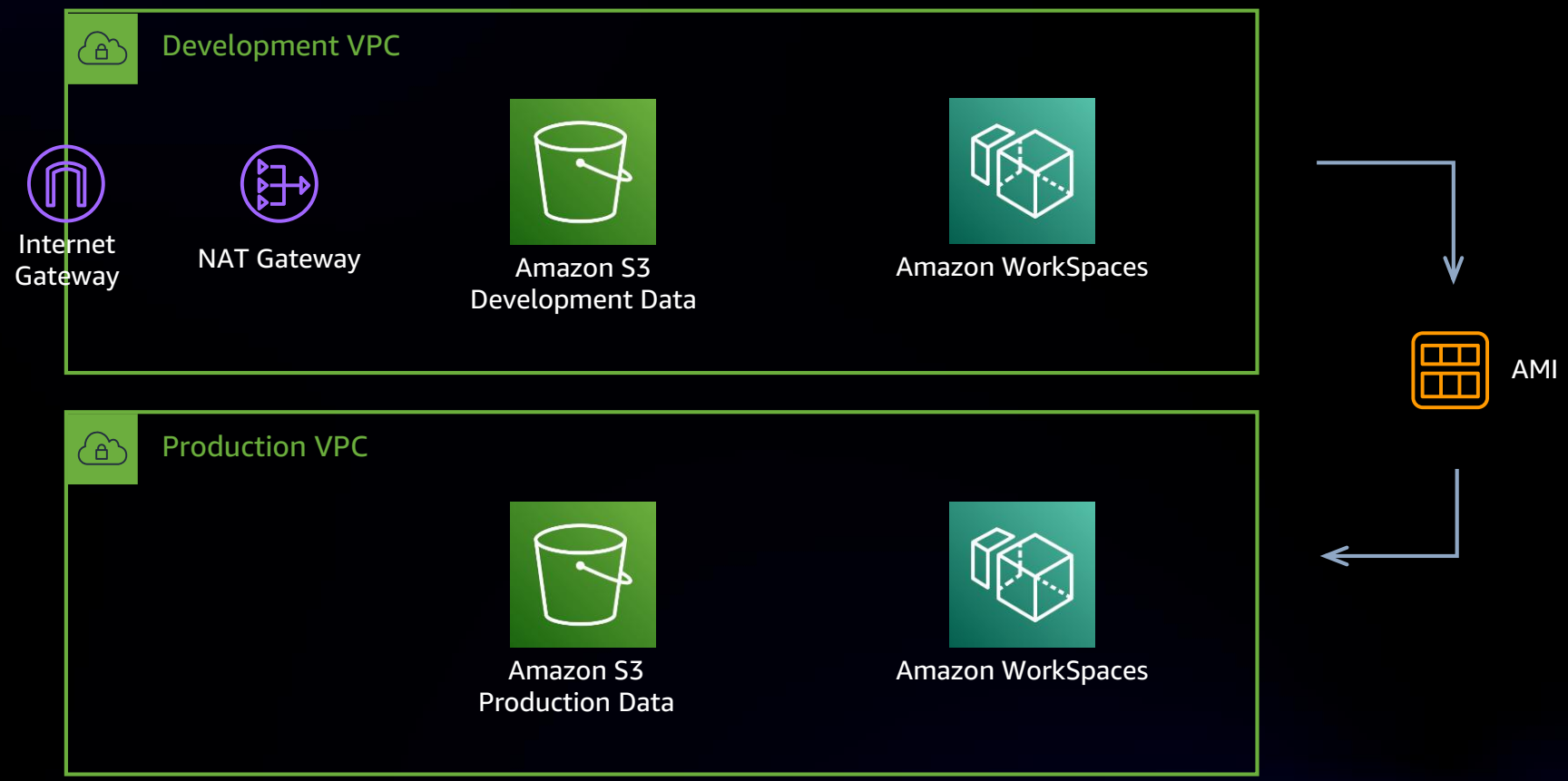
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

1. The 4-buckets architecture



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

2. Data science studio



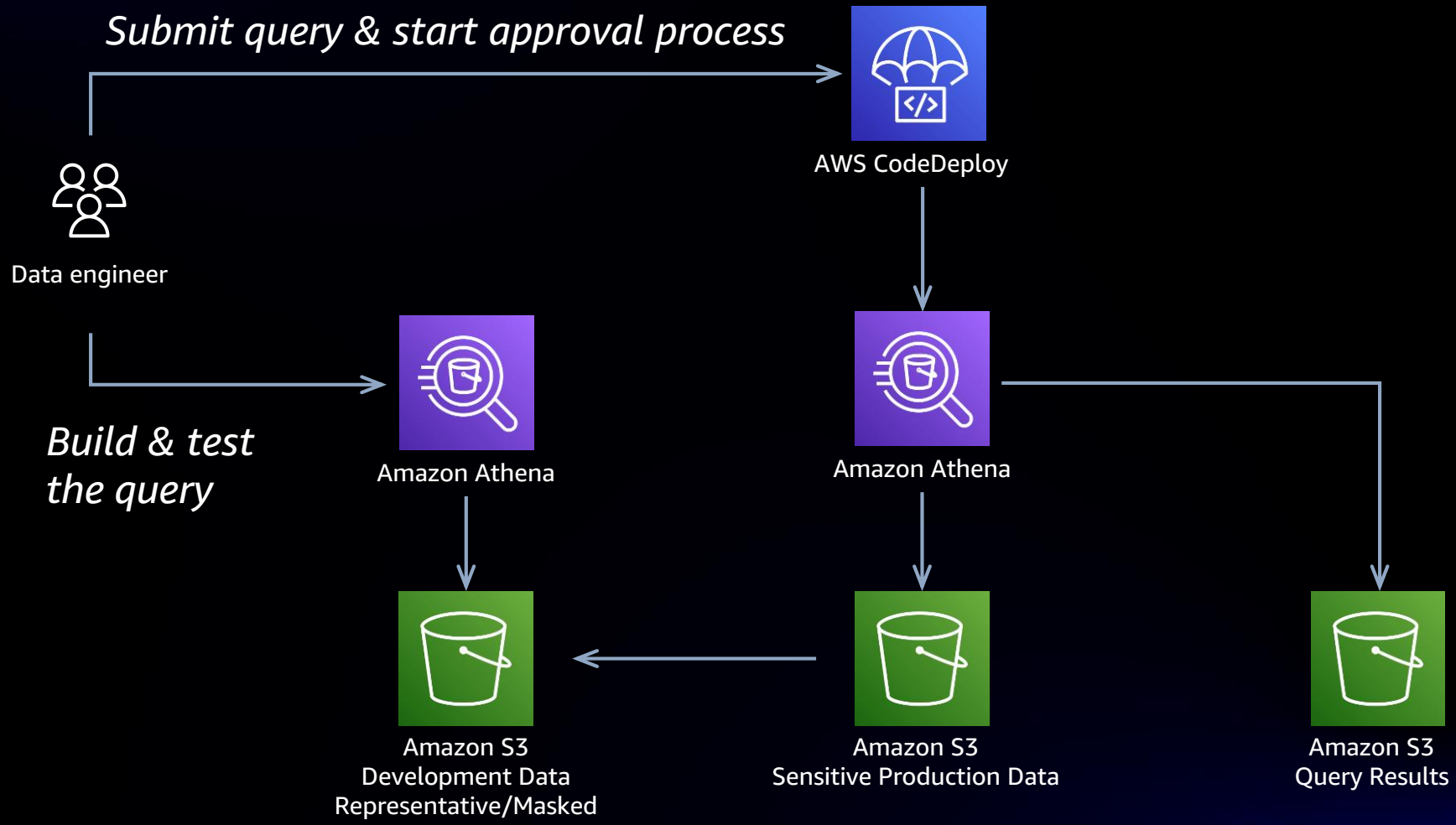
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

3. Encrypt all data

```
{
  "Version": "2012-10-17",
  "Id": "PutObjPolicy",
  "Statement": [
    {
    "Sid": "DenyIncorrectEncryptionHeader",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::<bucket_name>/*",
    "Condition": {
      "StringNotEquals": {
        "s3:x-amz-server-side-encryption": "aws:kms"
      }
    }
  ]
}
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4. Using a query pipeline



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11 best practices

Security

1. Use 4 buckets
2. Encrypt all data
3. Data science studio
4. Using a query pipeline

Access control

5. Infrastructure as code
6. Break-glass procedure
7. Working with metadata
8. Fine-grained access control

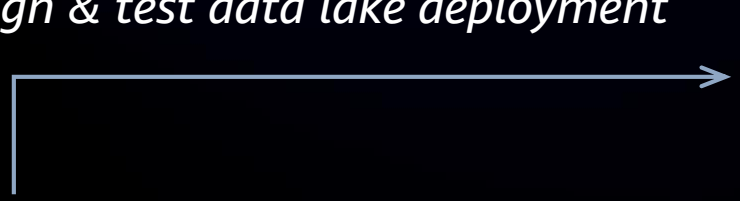
Compliance

9. Deleting data in the data lake
10. Detect violations with Amazon Macie
11. Data compliance dashboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

5. Infrastructure as code

Design & test data lake deployment





AWS Lake Formation



IAM



Development data



Test users



Data lake admins



Deploy with 4-eyes principle



AWS CodeDeploy



AWS CloudFormation



IAM



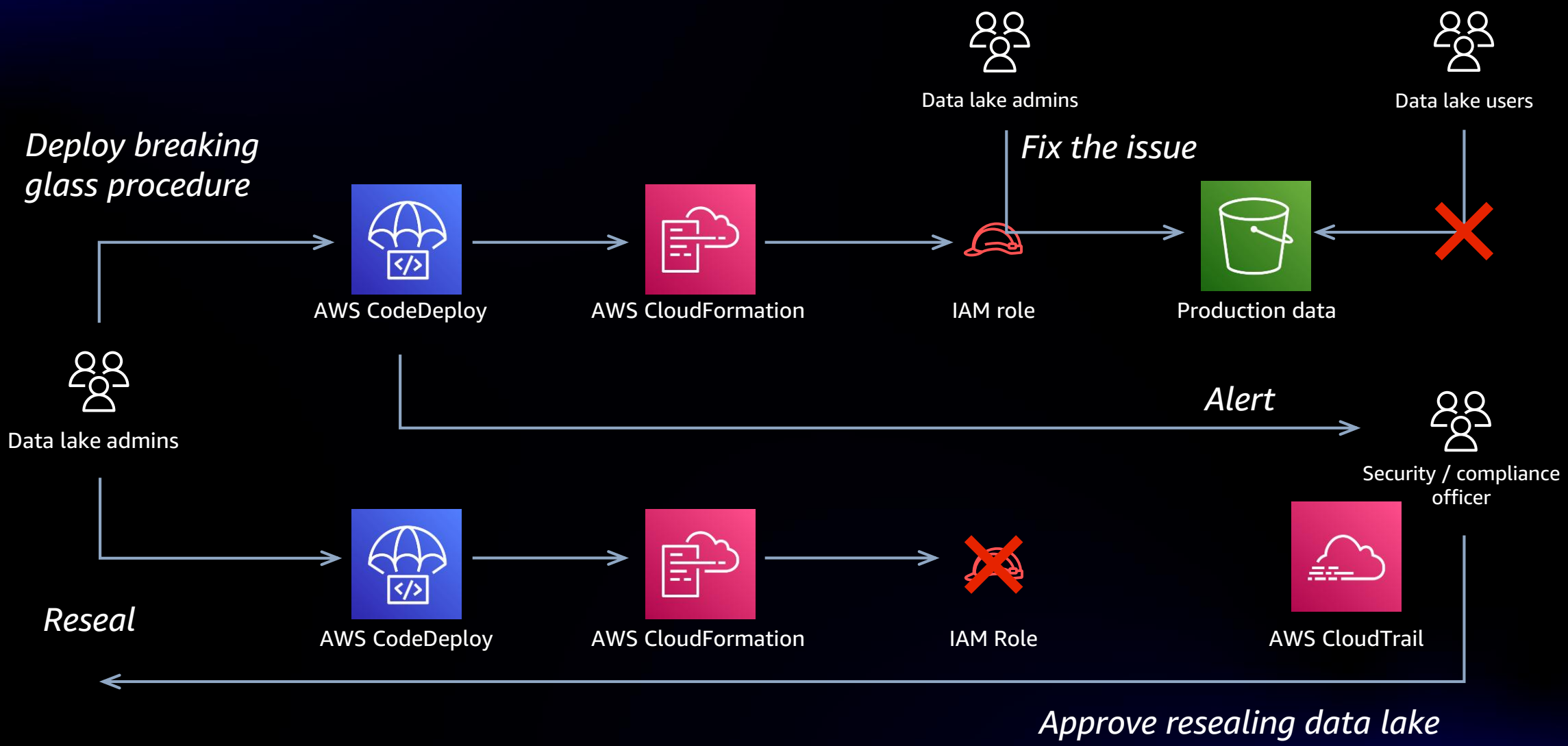
Production data



Production users

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

6. Break-glass procedure



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

7. Working with metadata

ID	Name	Location
1	Werner	Amsterdam
2	Andy	Seattle
3	Jeff	Seattle

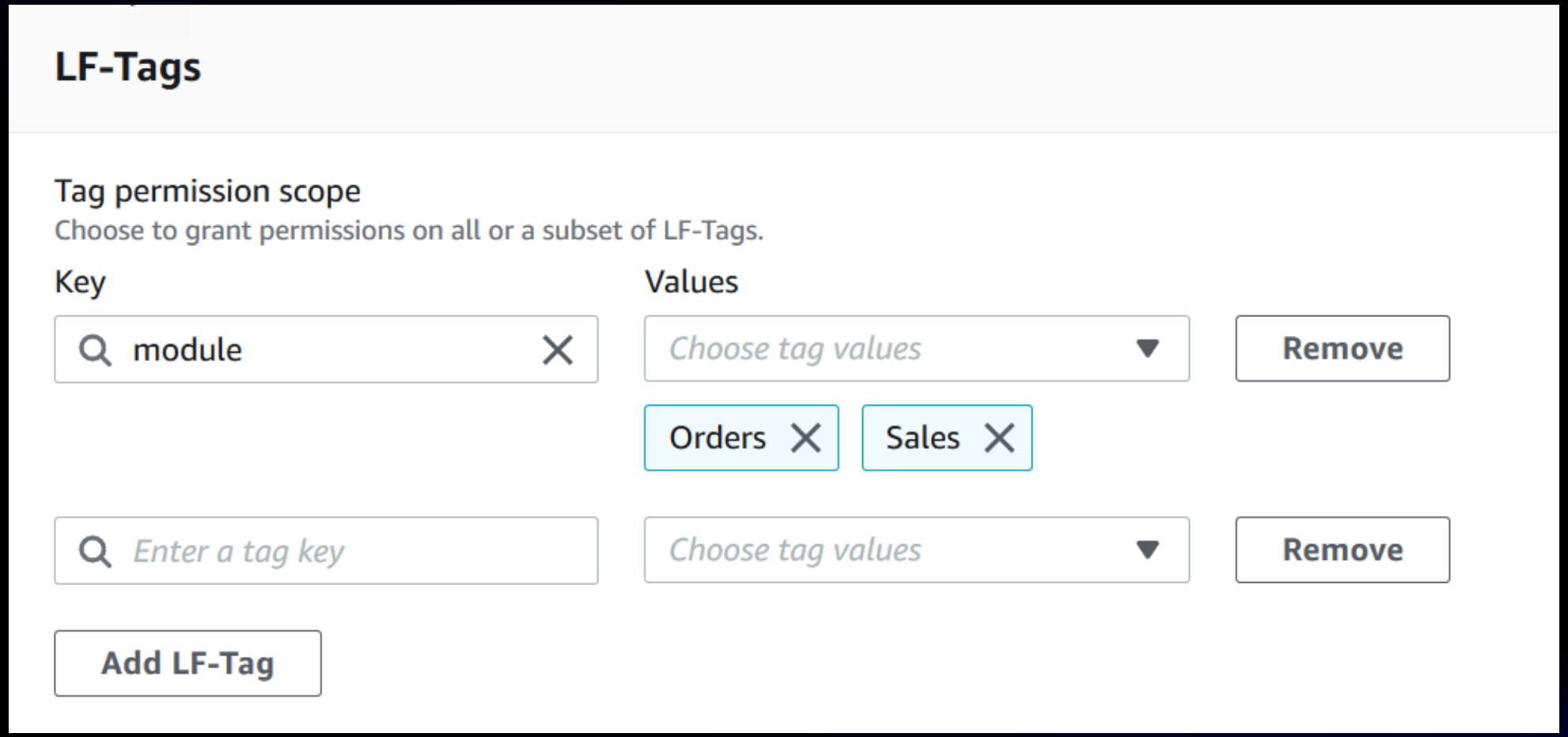
PII: True

ID	Mortgage	Location	PayGrade
1	500,000	Amsterdam	1
2	300,000	Seattle	2
3	1,000,000	Seattle	1

Domain: Sales

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. Fine-grained access control



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

8. Fine-grained access control in code

```
data "aws_lakeformation_permissions" "test" {
  principal = aws_iam_role.workflow_role.arn

  data_location {
    arn = aws_lakeformation_resource.test.arn
  }
}
```

```
Type: AWS::LakeFormation::Permissions
Properties:
  DataLakePrincipal:
    DataLakePrincipal
  Permissions:
    - String
  PermissionsWithGrantOption:
    - String
  Resource:
    Resource
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11 best practices

Security

1. Use 4 buckets
2. Encrypt all data
3. Data science studio
4. Using a query pipeline

Access control

5. Infrastructure as code
6. Break-glass procedure
7. Working with metadata
8. Fine-grained access control

Compliance

9. Deleting data in the data lake
10. Detect violations with Amazon Macie
11. Data compliance dashboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Deleting data in a data lake

Import data

ID	Name	Location
1	Werner	Amsterdam
2	Andy	Seattle
3	Jeff	Seattle

Hash table

ID	Hash
1	Hash(Werner, Amsterdam)
2	Hash(Andy, Seattle)
3	Hash(Jeff, Seattle)

Deleted data

ID	Hash

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Deleting data in a data lake

Import data

ID	Name	Location
1	Werner	Amsterdam
2	Andy	Seattle
3	Jeff	Seattle

Hash table

ID	Hash	Import
1	234j5lk3j45lk34j5	Approved
2	4h53ljkh53j45h34j	Approved
3	hj465lk45h6lk456h	Approved

Deleted data

ID	Hash

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Deleting data in a data lake

Import data

ID	Name	Location
1	Werner	Amsterdam
2		
3	Jeff	Seattle

Hash table

ID	Hash	Import
1	234j5lk3j45lk34j5	Approved
2	4h53ljkh53j45h34j	Not Imported
3	hj465lk45h6lk456h	Approved

Deleted data

ID	Hash
2	4h53ljkh53j45h34j

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

9. Deleting data in a data lake

Import data

ID	Name	Location
1	Werner	Amsterdam
~~2~~	~~Andy~~	~~Seattle~~
3	Jeff	Seattle

Hash table

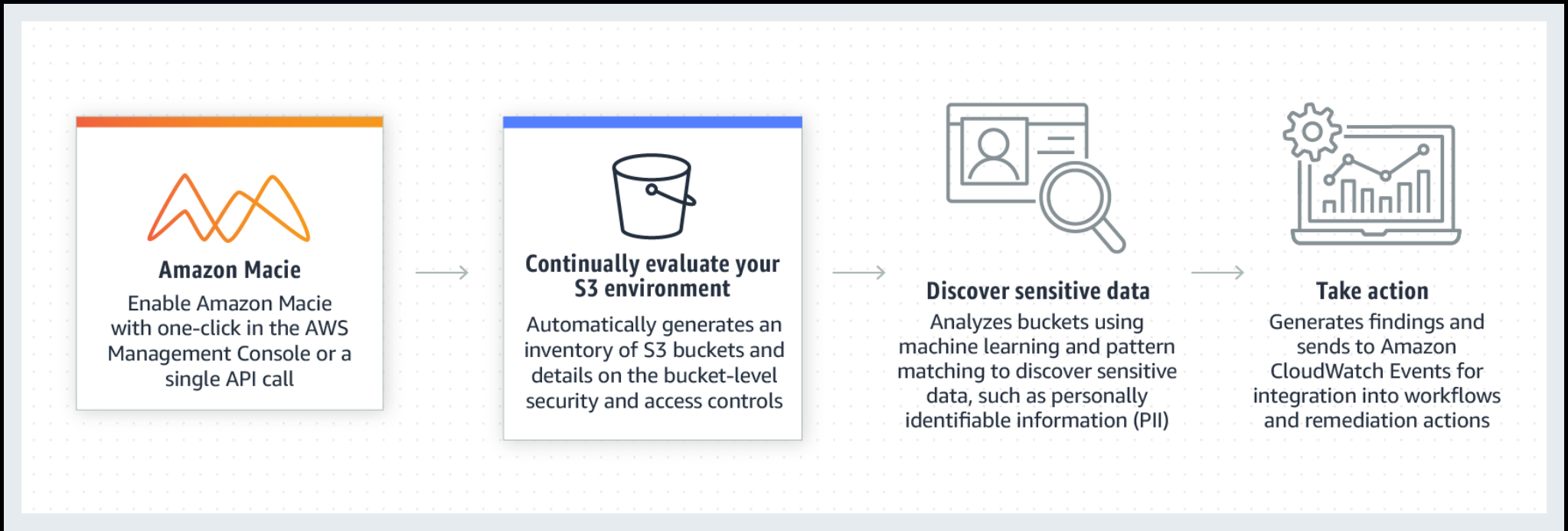
ID	Hash	Import
1	234j5lk3j45lk34j5	Approved
2	4h53ljkh53j45h34j	Import Denied
3	hj465lk45h6lk456h	Approved

Deleted data

ID	Hash
2	4h53ljkh53j45h34j

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

10. Detect PII with Macie



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11. Data compliance dashboard

1

Break-glass procedures used

65

Amazon Macie violations

23

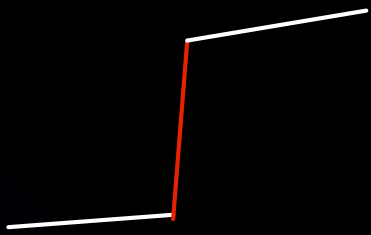
Records without mandatory tags

83

Import violations



Deployments



Data lake size

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

11 Best practices

Security

1. Use 4 buckets
2. Encrypt all data
3. Data science studio
4. Using a query pipeline

Access control

5. Infrastructure as code
6. Break-glass procedure
7. Working with metadata
8. Fine-grained access control

Compliance

9. Deleting data in the data lake
10. Detect violations with Amazon Macie
11. Data compliance dashboard

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next steps

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recommended resources

1

Analytics Lens – AWS Well-Architected Framework

https://docs.aws.amazon.com/wellarchitected/latest/analytics-lens/wellarchitected-analytics-lens.pdf

2

Whitepaper: Cloud Native Data Virtualization on AWS

https://d1.awsstatic.com/whitepapers/cloud-native-data-virtualization-on-aws.pdf?did=wp_card&trk=wp_card

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Want to learn more?

ANT311
Build a data mesh with AWS Lake Formation and AWS Glue

ANT320
Building data lakes and sharing data with AWS Lake Formation

ANT312
Building a secure, modern data architecture with AWS analytics

ARC321
Modern data architectures using the AWS Well-Architected Analytics Lens

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Martijn van Dongen

bio.link/mvd
awsug.nl
schubergphilis.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.