
AWS
re:Invent
DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

COP326

Unlocking business insights with AWS Config, featuring Itaú Unibanco

Matheus Arrais
Sr. WW CloudOps SA
AWS

Guilherme Greco
Principal SA
AWS

Thiago Morais
Head of Cloud Platform
Itaú Unibanco

Have you used inventory data to make business decisions?

What we will cover today

Customer challenges

How to address with AWS

Itaú Unibanco use case

Key takeaways

Resources

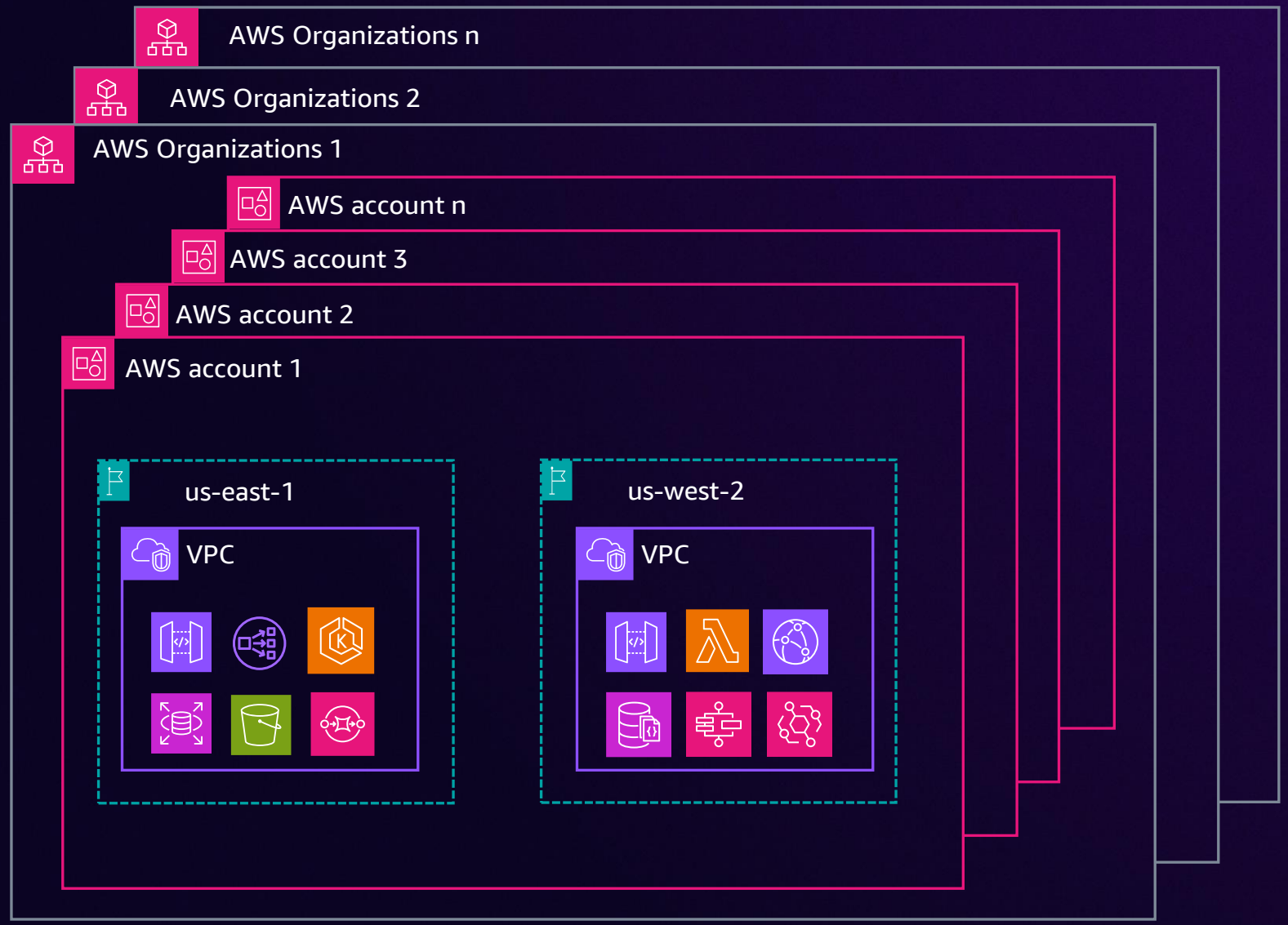
Our customers' challenges

Dynamic landscape

Continuous governance and compliance

Environment changes

AnyCompany architecture



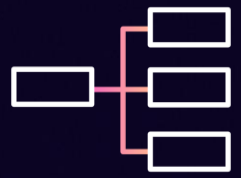
AnyCompany architecture

How many resources do we have?

How many resources belong to specific business unit or product?

How many resources follow our architectural pattern?

AnyCompany architecture



Resource inventory

Compliance management

Leverage resource configuration data for business insights

Historical data

Services relationship

Cost optimization

How are customers solving these?

Resource configuration

- Relationship
- Size
- Network
- Security


```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2024-10-23T15:42:47.538Z",
  "configurationItemStatus": "OK",
  "configurationStateId": "1729698167538",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:ec2:us-east-1:111111111111:security-group/sg-0a7b3c17a8d453579",
  "resourceType": "AWS::EC2::SecurityGroup",
  "resourceId": "sg-0a7b3c17a8d453579",
  "resourceName": "demo-riv2024",
  "awsRegion": "us-east-1",
  "availabilityZone": "Not Applicable",
  "tags": {},
  "relatedEvents": [],
  "relationships": [ …
  ],
  "configuration": { …
  },
  "supplementaryConfiguration": {}
}
```

How do I get started?

AWS Config

AWS Config

DISCOVER, ASSESS, AUDIT, AND REMEDIATE

Configuration changes occur in your AWS resources

AWS Config features

- Recording and normalization of the changes into a consistent format
- Rules and conformance packs evaluate compliance
- Aggregators to collect data from multiple accounts and Regions; advanced queries identify and retrieve data
- Remediate noncompliant resources by using AWS Systems Manager

Where you can send AWS Config data:

- AWS Config APIs and console
- Amazon SNS
- Amazon CloudWatch
- Amazon S3
- AWS CloudTrail Lake
- Amazon EventBridge

AWS Config recorder

How does AWS Config record resource configuration?

CONFIGURATION ITEM

Metadata

Attributes

Relationships

Current configuration


```
{
  "version": "1.3",
  "accountId": "111111111111",
  "configurationItemCaptureTime": "2024-10-09T20:23:16.211Z",
  "configurationItemStatus": "OK",
  "configurationStateId": "1728505396211",
  "configurationItemMD5Hash": "",
  "arn": "arn:aws:ec2:us-east-1:111111111111:security-group/sg-0a48e96701930b4c8",
  "resourceType": "AWS::EC2::SecurityGroup",
  "resourceId": "sg-0a48e96701930b4c8",
  "resourceName": "launch-wizard-1",
  "awsRegion": "us-east-1",
  "availabilityZone": "Not Applicable",
  "tags": {},
  "relatedEvents": [],
  "relationships": [
    {
      "resourceType": "AWS::EC2::Instance",
      "resourceId": "i-0d75106d1c993398f",
      "relationshipName": "Is associated with Instance"
    },
    {…
    },
    {…
    }
  ],
  "configuration": {…
  },
  "supplementaryConfiguration": {}
}
```

How does AWS Config record resource configuration?

CONFIGURATION ITEM

Metadata
Attributes
Relationships
Current configuration


```
"configuration": {
  "description": "Demo reinvent2024",
  "groupName": "demo-riv2024",
  "ipPermissions": [],
  "ownerId": "111111111111",
  "groupId": "sg-0a7b3c17a8d453579",
  "ipPermissionsEgress": [
    {
      "ipProtocol": "-1",
      "ipv6Ranges": [],
      "prefixListIds": [],
      "userIdGroupPairs": [],
      "ipv4Ranges": [
        {
          "cidrIp": "0.0.0.0/0"
        }
      ],
      "ipRanges": [
        "0.0.0.0/0"
      ]
    }
```

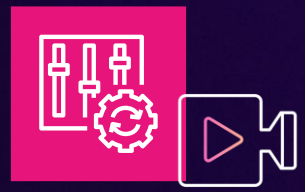
How does AWS Config record resource configuration?

CONFIGURATION ITEM

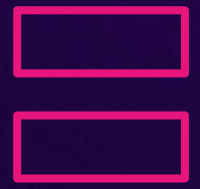
Configuration history

Configuration snapshot

Configuration item



Configuration recorder

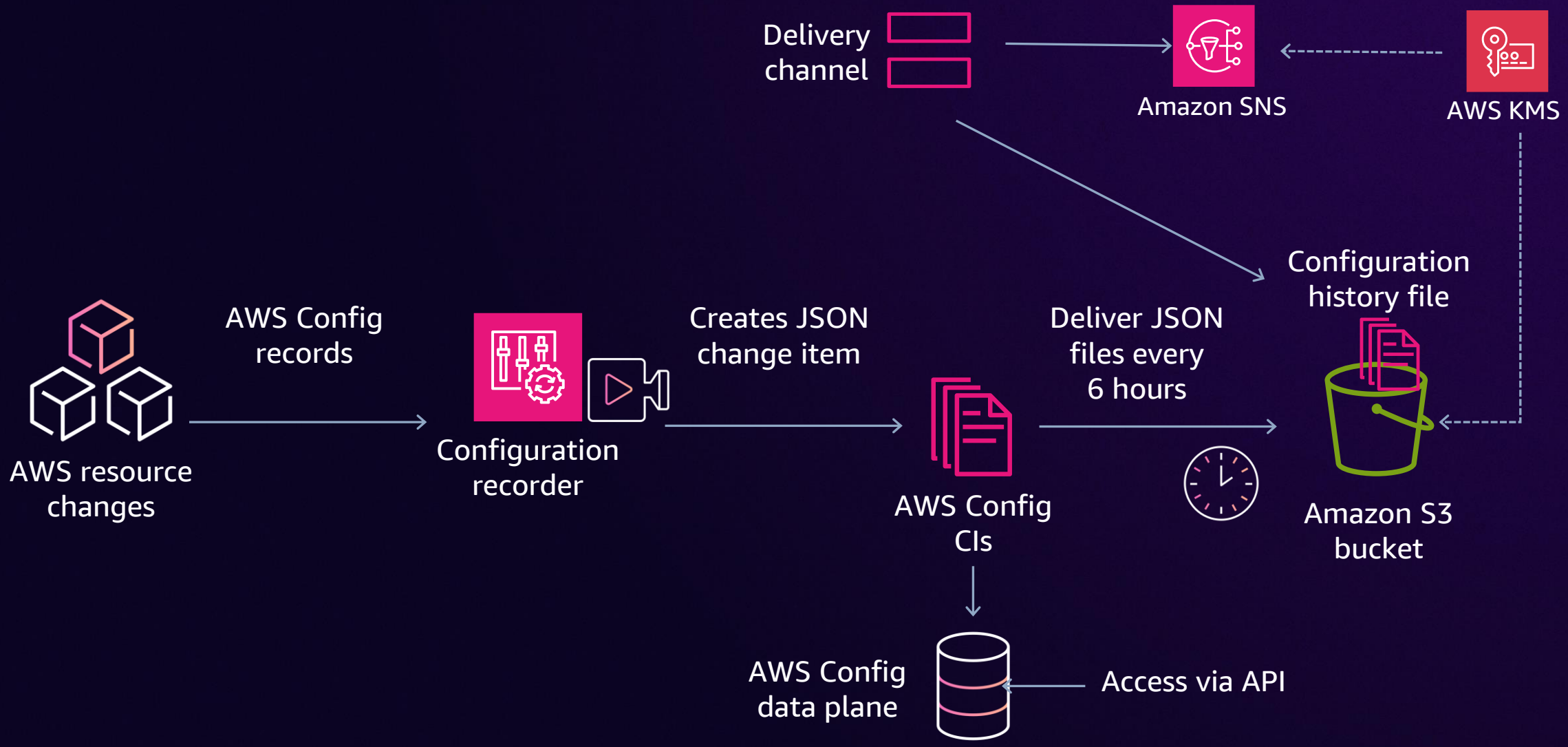


Delivery channel

AWS Config recorder

How can I keep track of my resource configuration?

AWS CONFIG RECORDER

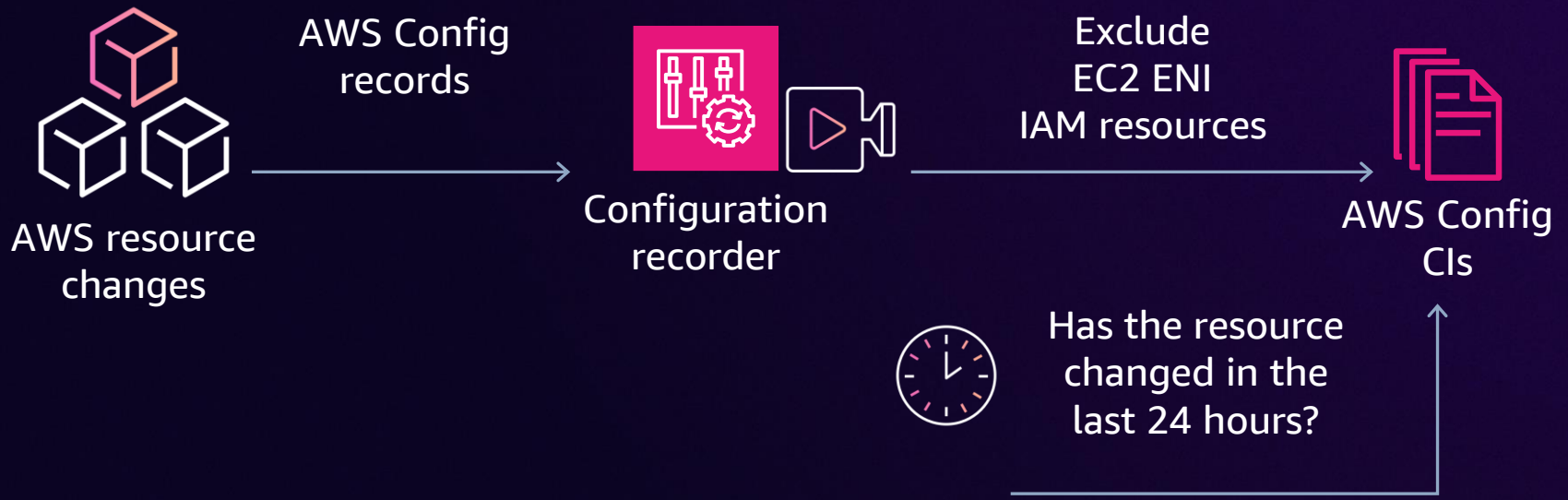


How do I know what's deployed right now?

AWS CONFIG RECORDER

Do I need to record everything?

AWS CONFIG RECORDER

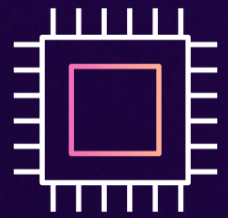


Can I record other things?

AWS CONFIG CUSTOM RESOURCES

Enrich configuration data

Add internal information

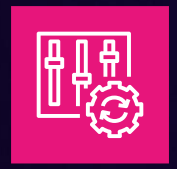


Custom and third-party resources

Can I record other things?

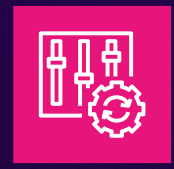
AWS CONFIG CUSTOM RESOURCES

Amazon EKS clusters



AWS::EKS::Cluster

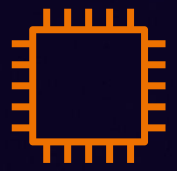
AnyCompany::EKS::Cluster



AnyCompany::X::Y

EKS cluster configurations

Enriched configurations
Managed node groups
Upgrade insights

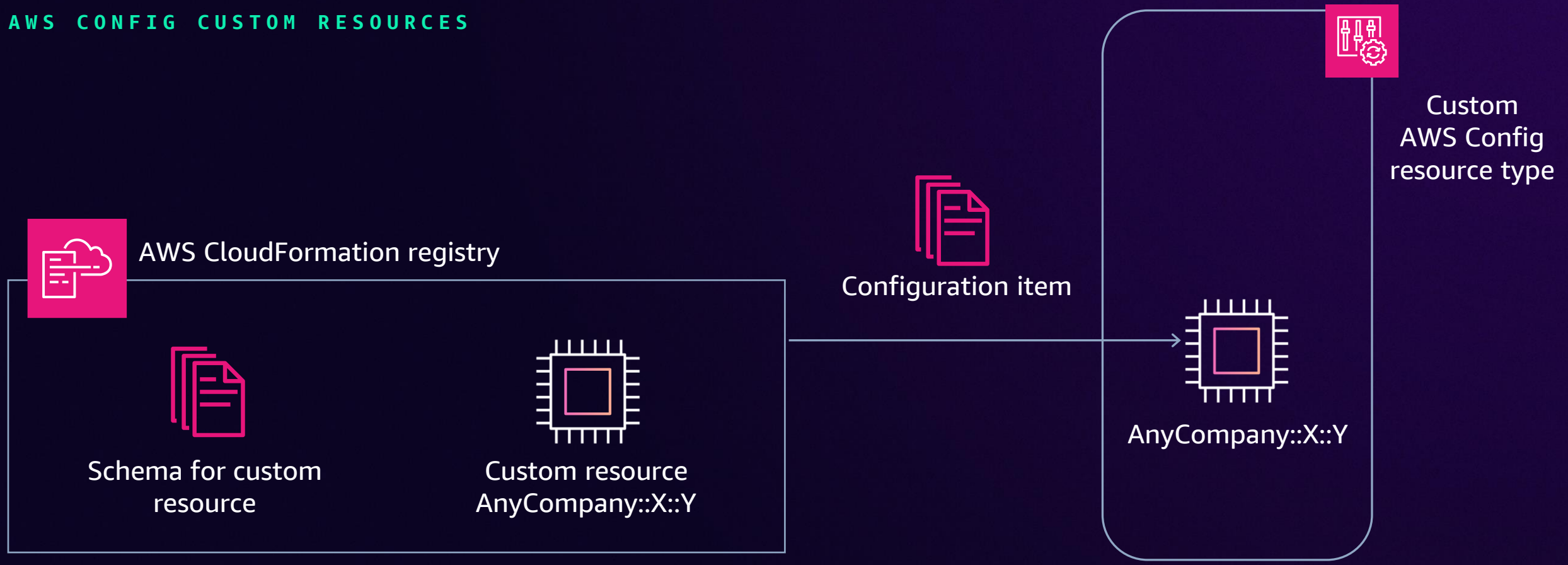


Third party

Third-party configurations

How does AWS Config support customized resources?

AWS CONFIG CUSTOM RESOURCES



How to check resource compliance

Is this good or bad?

October 17, 2024 3:11 PM

Security group

fromPort: 22
cidrIpv4: 192.168.1.23/32

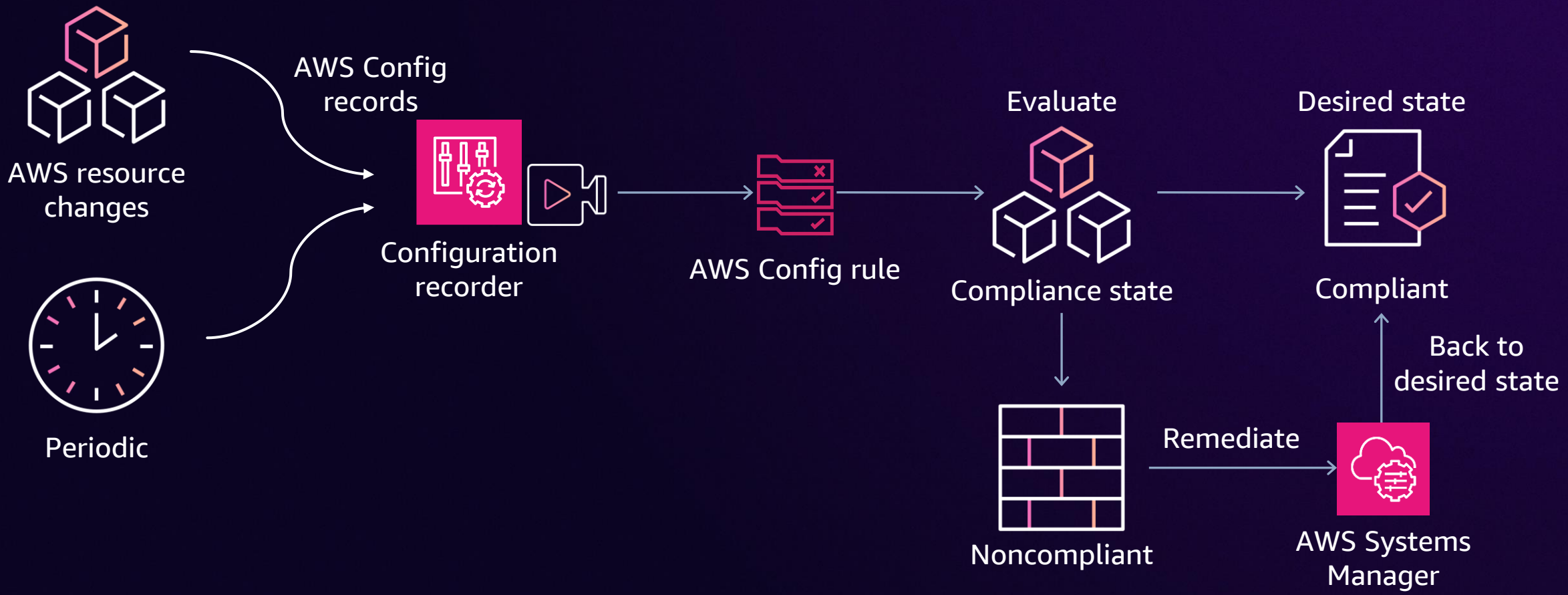
October 17, 2024 3:23 PM

Security group

fromPort: 22
cidrIpv4: 0.0.0.0/0

AWS Config rules

How do AWS Config rules work?



Types of AWS Config rules

Managed rules

- Defined and managed by AWS
- Require minimal (or no) configuration
- 500+ rules **ready** to use

restricted-ssh

required-tags

rds-storage-encrypted

Custom rules

- Authored and maintained by you
- Use AWS Lambda, AWS CloudFormation Guard, or RDK

anycompany-rule1

anycompany-rule2

anycompany-rule3

AWS Config custom rules via AWS Lambda

Python code that evaluates if an instance type is in the allowed list type for compliance

```
import boto3
import json
ASSUME_ROLE_MODE = False
def get_client(service, event):
    if not ASSUME_ROLE_MODE:
        return boto3.client(service)
   credentials = get_assume_role_credentials(event["executionRoleArn"])
    return boto3.client(service, aws_access_key_id=credentials['AccessKeyId'],
                        aws_secret_access_key=credentials['SecretAccessKey'],
                        aws_session_token=credentials['SessionToken']
                       )
# Helper function used to validate input
def check_defined(reference, reference_name):
    if not reference:
        raise Exception('Error: ', reference_name, 'is not defined')
    return reference
# Check whether the message is OversizedConfigurationItemChangeNotification or not
def is_oversized_changed_notification(message_type):
   check_defined(message_type, 'messageType')
    return message_type == 'OversizedConfigurationItemChangeNotification'
# Get configurationItem using getResourceConfigHistory API
# in case of OversizedConfigurationItemChangeNotification
def get_configuration(resource_type, resource_id, configuration_capture_time):
   result = AWS_CONFIG_CLIENT.get_resource_config_history(
       resourceType=resource_type,
       resourceId=resource_id,
       laterTime=configuration_capture_time,
       limit=1)
   configurationItem = result['configurationItems'][0]
    return convert_api_configuration(configurationItem)
# Convert from the API model to the original invocation model
def convert_api_configuration(configurationItem):
    for k, v in configurationItem.items():
...
```

AWS Config custom rules via cfn-guard

Author AWS Config rule using CloudFormation Guard (domain-specific language)

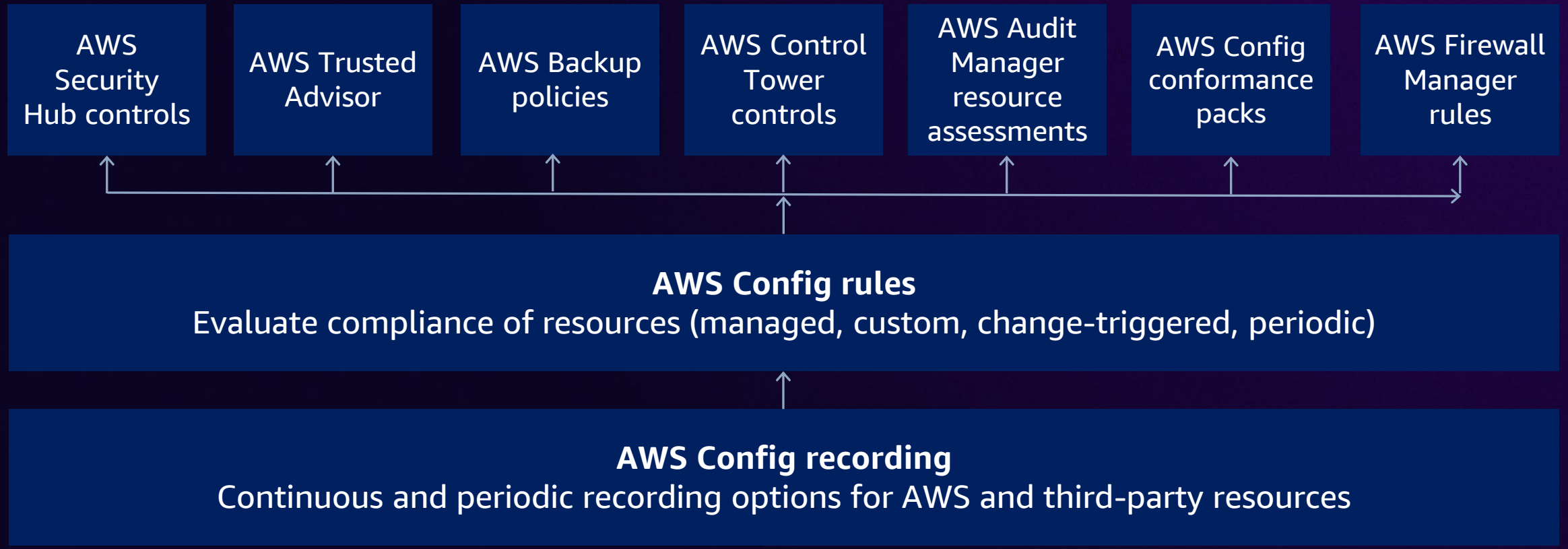
Benefits

- Less error handling – permissions, oversized CIs, deleted CIs
- Assuming roles, PutEval calls, and infrastructure management not required
- Ease of authoring and deployment
- Policy as code

GitHub Config rules samples

AWS Config

CORE FOR COMPLIANCE EVALUATIONS

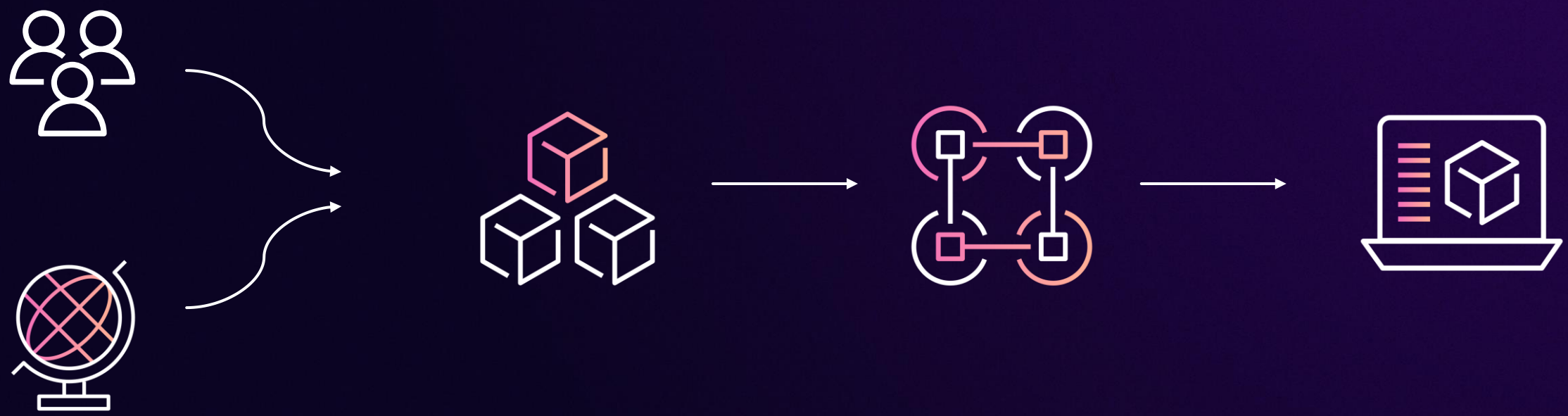


How to consolidate this data

AWS Config aggregator

Consolidating my resource configuration data

AWS CONFIG AGGREGATOR



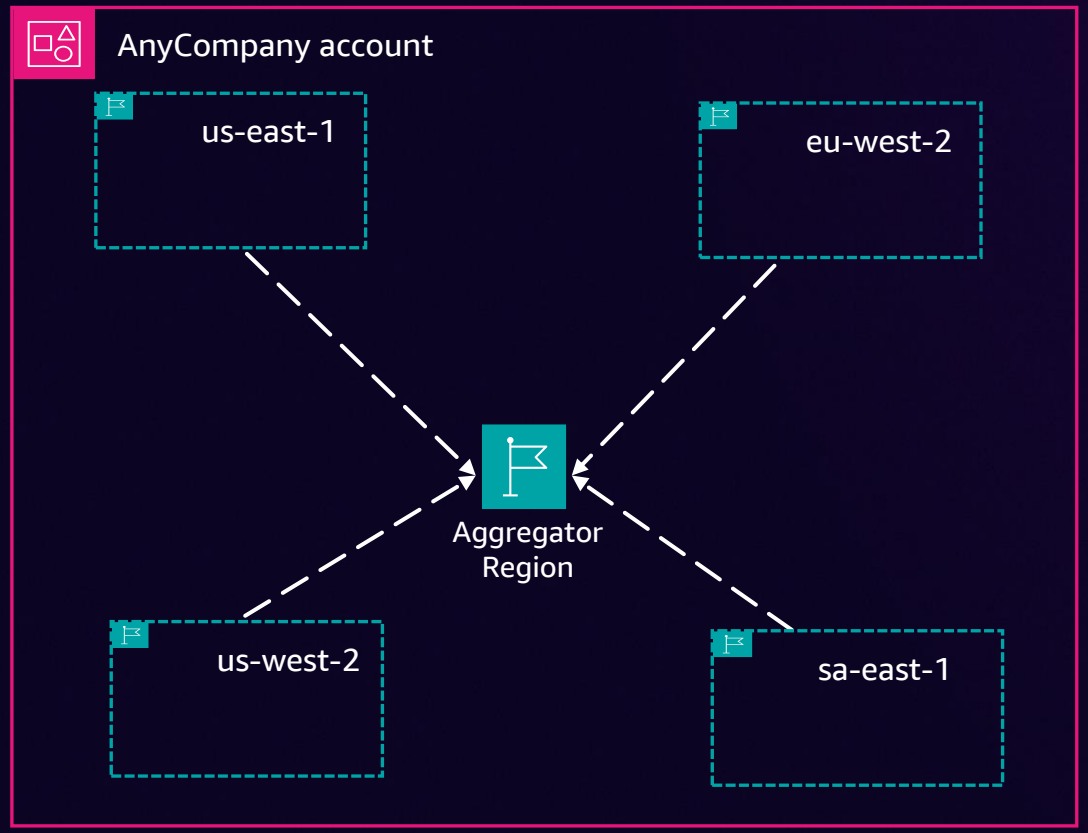
Accounts and Regions

AWS Config data

Aggregator

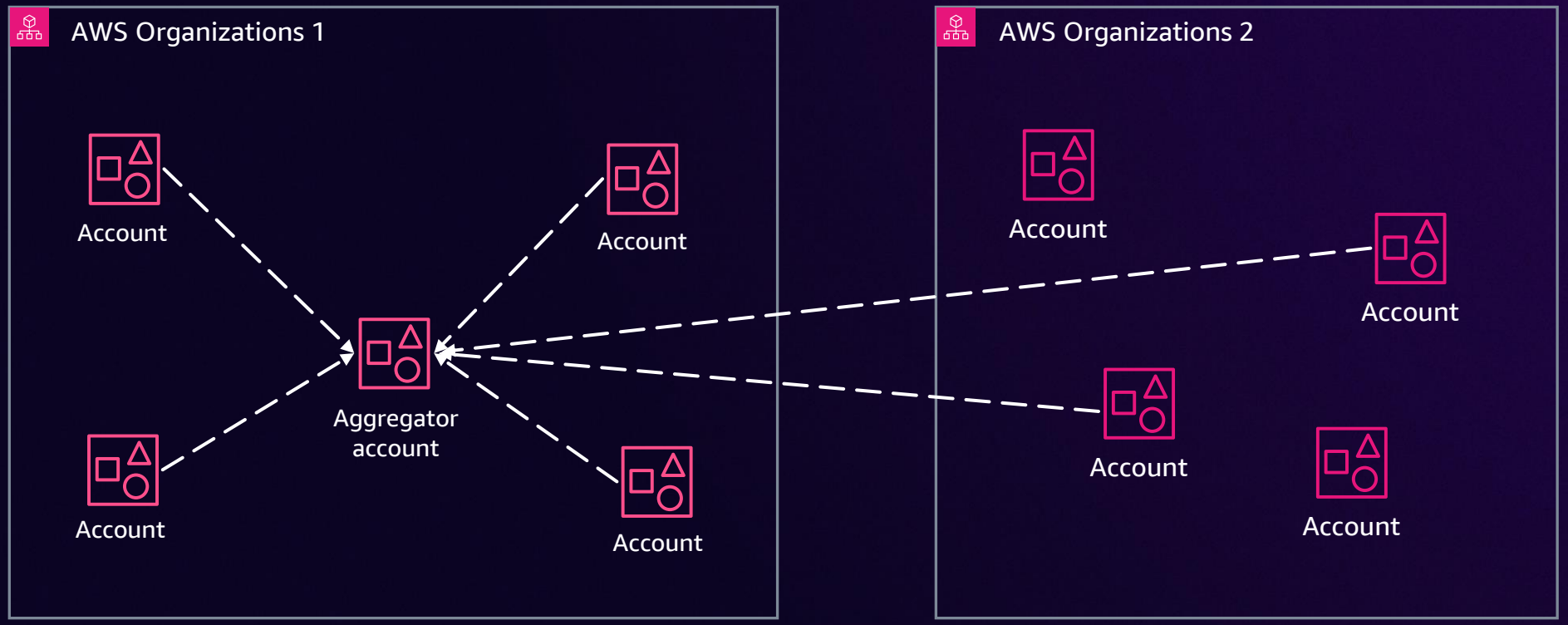
Aggregated view

Individual account aggregator



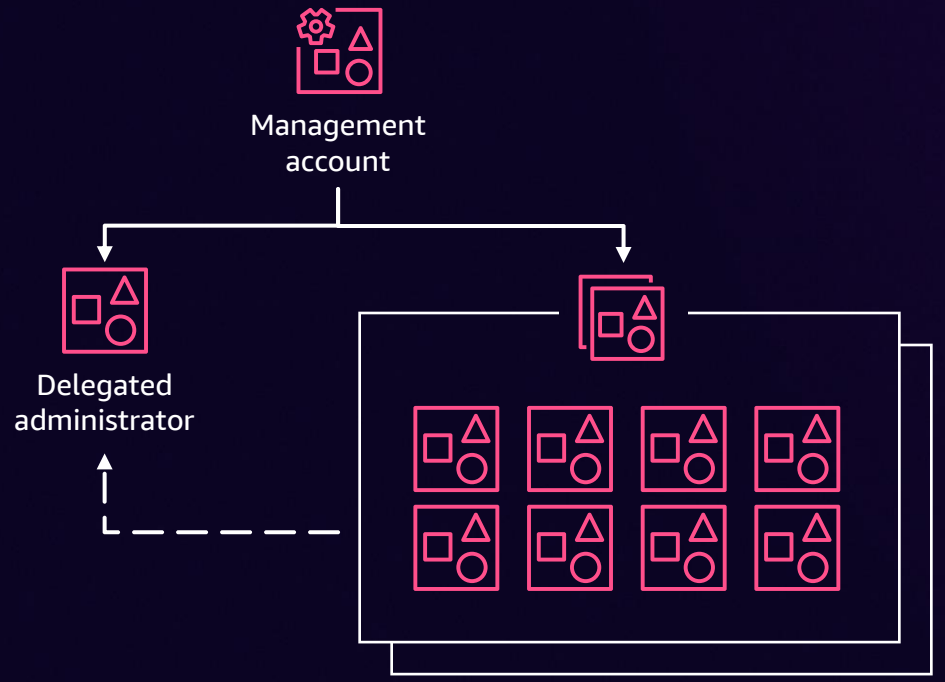
Centralize multi-Region data in a central Region

Individual account aggregator



External accounts

Organization aggregator



Centralize data across your AWS Organizations in a delegated central account

AWS Config advanced query

Gathering business insights

AWS CONFIG ADVANCED QUERY

Multi-account and multi-Region aggregation of resources

Perform ad hoc and property-based queries to query the current resource state

Security and operational intelligence

Cost optimization

AWS Config: Demo

Ok. But it’s not only about the resource configuration . . .

Ok. But it’s not only about the resource configuration . . .

It’s about the business value

Itaú Unibanco

Itaú Unibanco

A universal bank with 100 years of history; largest financial institution in Latin America[1]

Market value[1]
US$ 60.1B

Recurring ROE[3]
22.7%

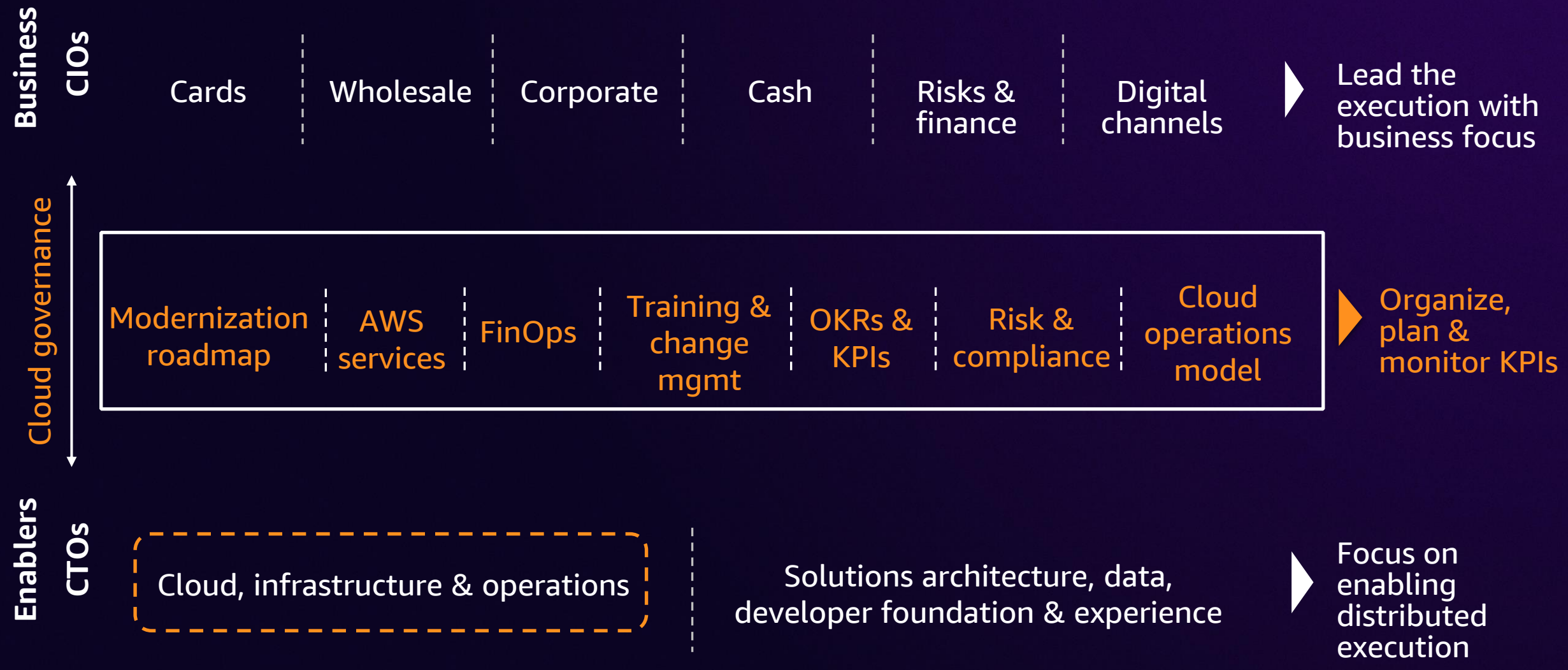
Most valuable brand in Latin America[4]
US$ 8.4B

Total assets[2]
US$ 3.009B

96.8K employees in Brazil and overseas[2]

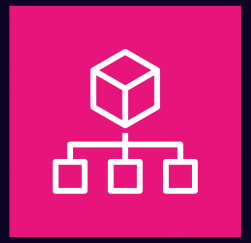
(1) Market value in August 02, 2024; (2) In July 30, 2024;
(2) (3) In the 2nd quarter of 2024; (4) Brand Finance – Latin America 500 2024

How we are organized

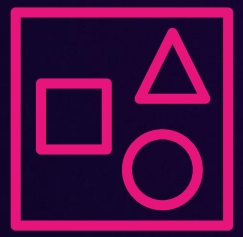


Itaú’s cloud footprint

Itaú's multi-account environment



5 AWS Organizations



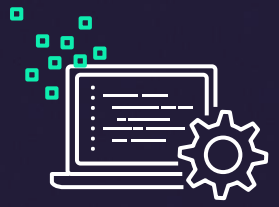
9k+ AWS accounts

40M+ configuration items

100+ communities
300 release trains
2,700+ squads

5,700+ business services

What were our challenges?

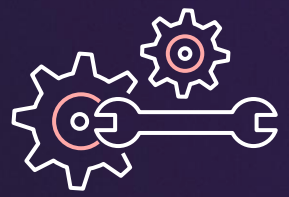


Visibility

Visibility of configuration changes over AWS resources using AWS Config

Multi-org platform in a single panel

Cloud metadata shared in data mesh



Monitoring

Database, container, and compute lifecycles

Integrated indicators of quality, resiliency, security, and cost

Planning

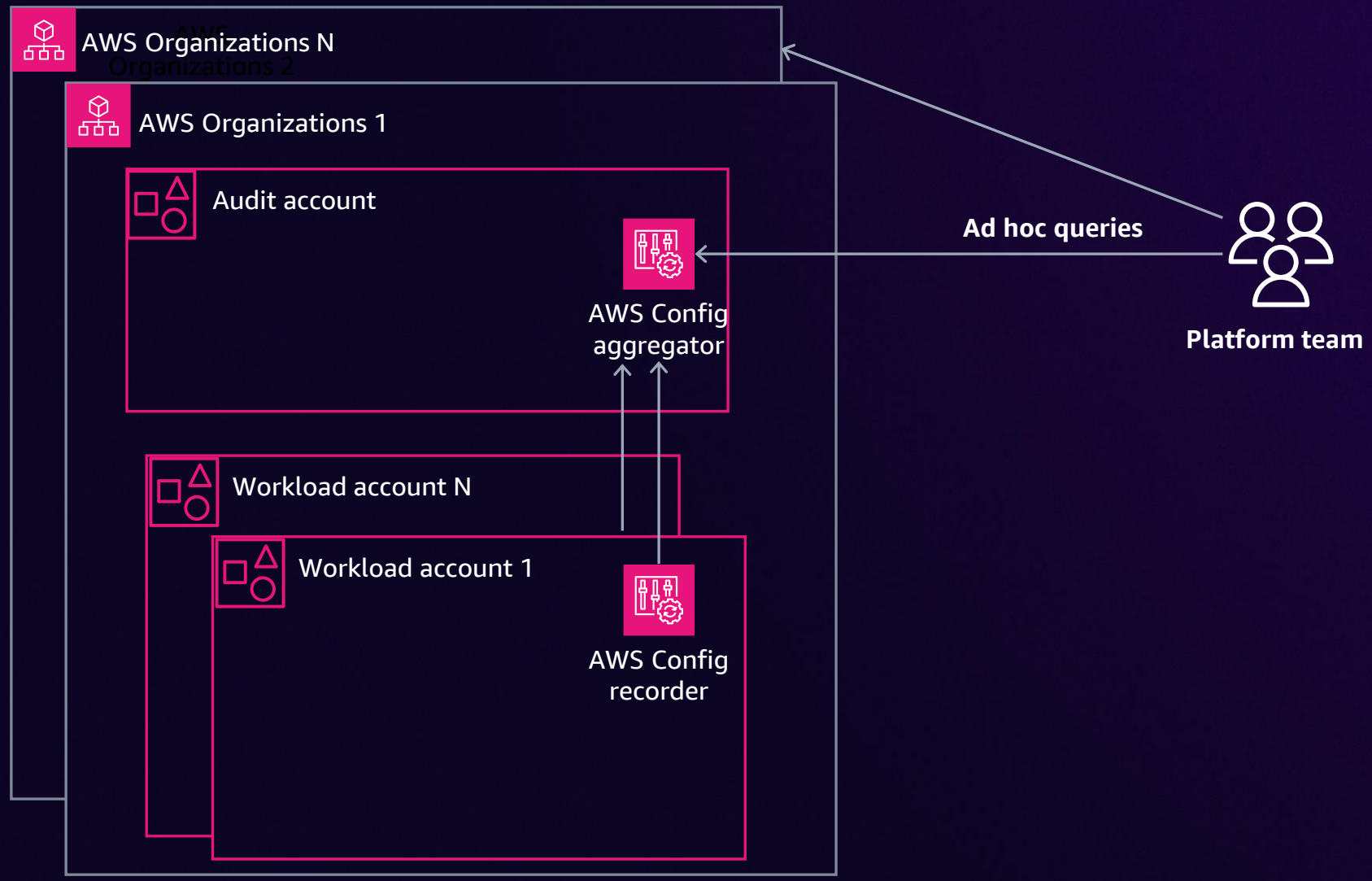
Assess compliance policies and scores

Data-driven decisions

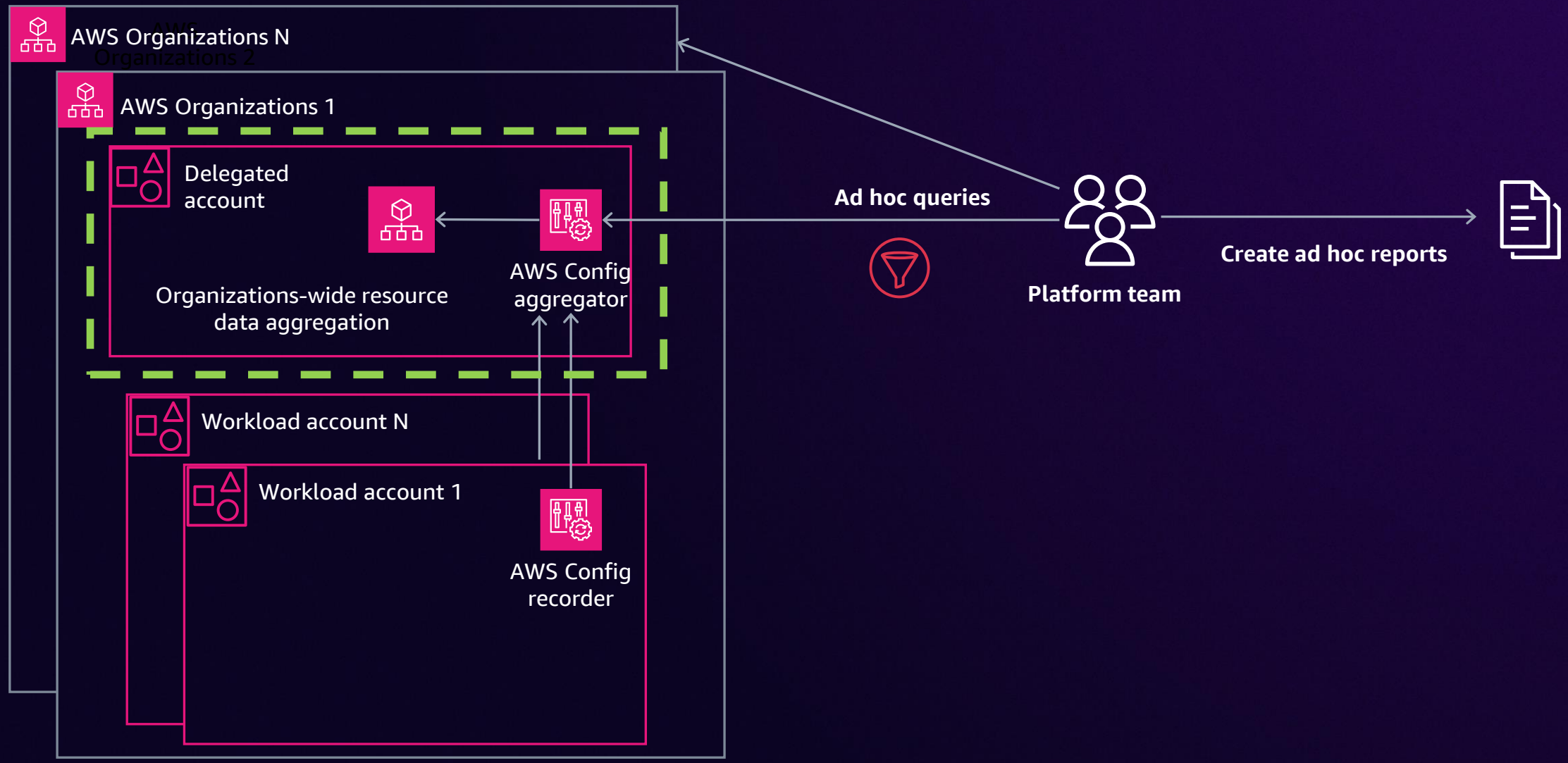
Proactive remediation of backlevel

Itaú’s cloud metadata journey with AWS Config

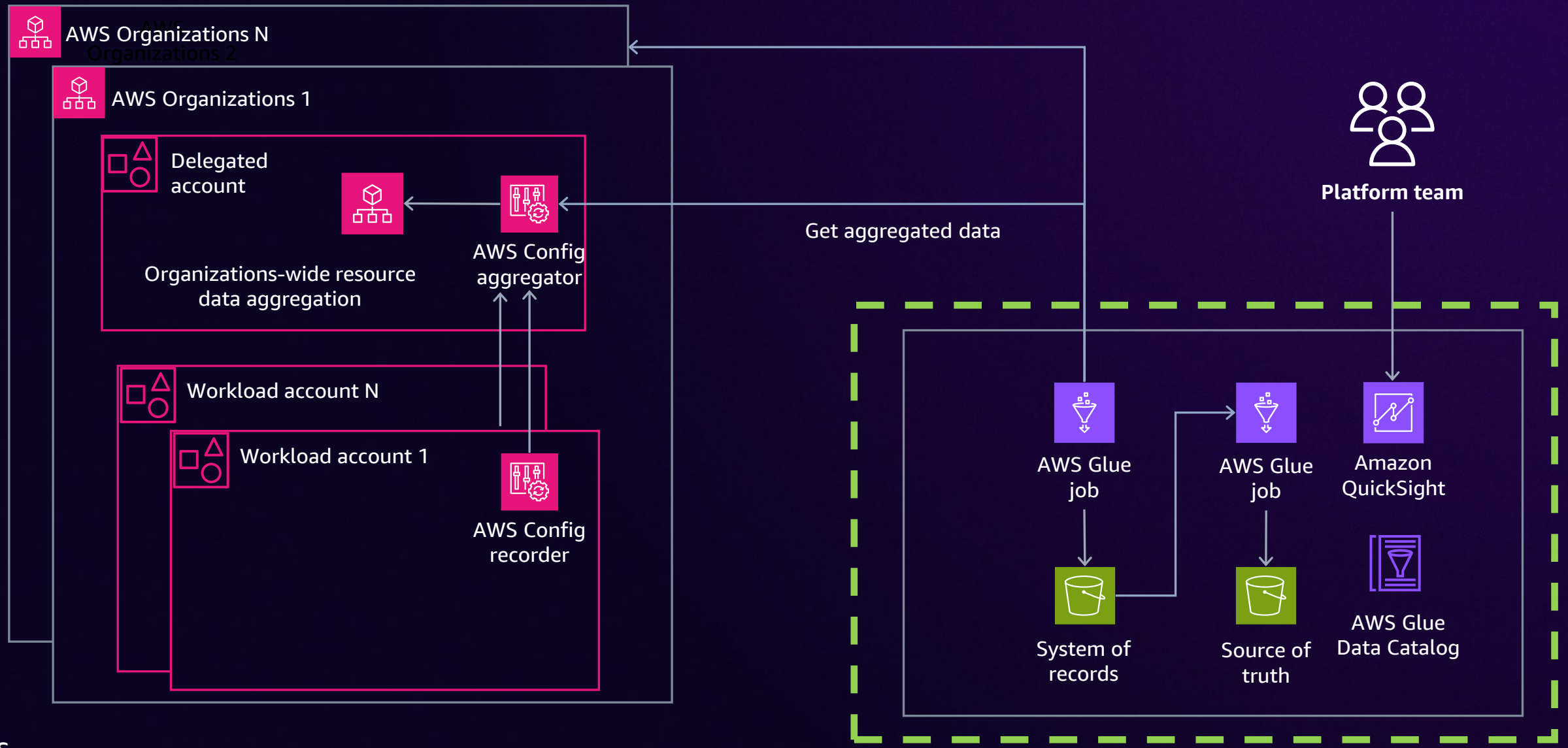
First steps



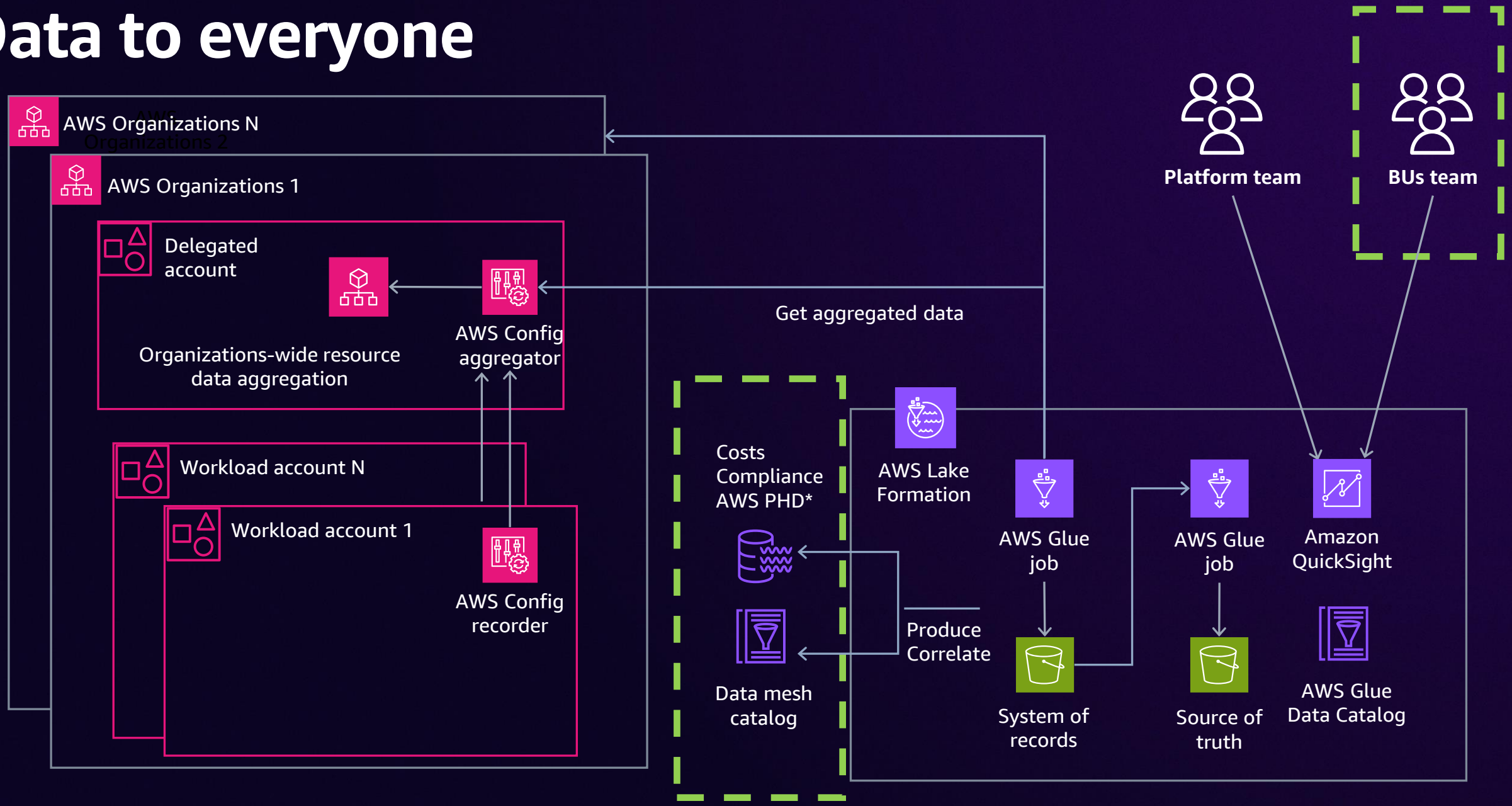
Bringing best practices



Moving to a data pipeline strategy

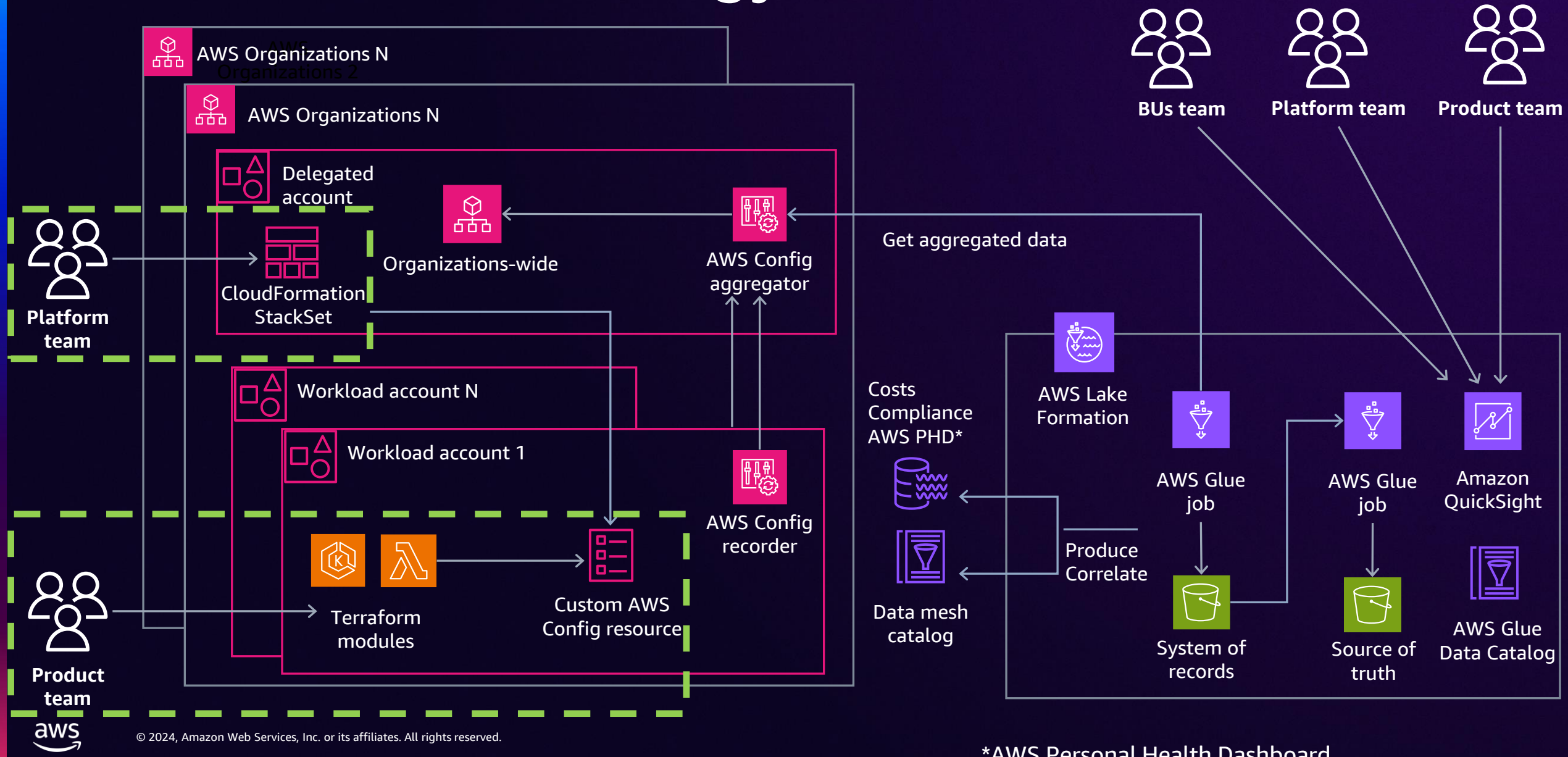


Data to everyone



*AWS Personal Health Dashboard

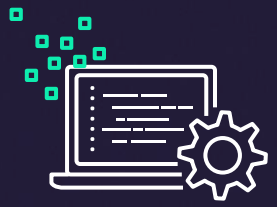
Data enrichment strategy



*AWS Personal Health Dashboard

Itaú’s demo

What is next?

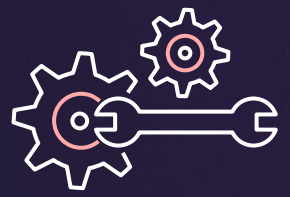


Visibility

Configuration changes over AWS resources using AWS Config

Multi-org platform in a single panel

Cloud metadata shared in data mesh



Monitoring

Database, container, and compute lifecycles

Integrated indicators of quality, resiliency, security, and cost

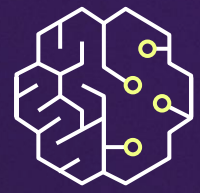
Assess compliance

Planning

Policies and scores

Data-driven decisions

Proactive remediation of backlevel



Augmented insights

Storytelling and insights powered by gen AI

Platform automation with domain-specific agents

JOURNEY

Present → Future

Wrapping up

AWS Config recap

Resource inventory

Compliance and auditing

Troubleshoot and remediate

Key takeaways

01 Use AWS Config for resource configuration data at scale

02 Extracts resource configuration value

03 Make data-driven decisions

Resources

Getting Started with AWS Config

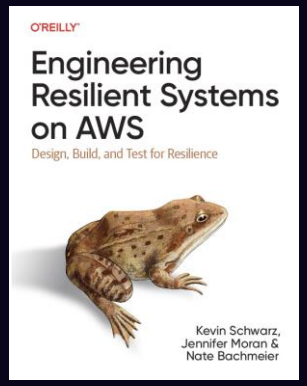
YouTube – Track configuration changes at scale using AWS Config

Blog post – Set up an organization-wide aggregator

Cloud Ops Kiosks

Cloud Operations | Observability | Governance & Compliance | Resilience | Cloud Financial Management

VR EXPERIENCE



BOOK GIVEAWAYS

SWAG

MEET US AT THE KIOSKS IN THE AWS VILLAGE

Thank you!

Please complete the session survey in the mobile app

Matheus Arrais
LinkedIn: @matheusarrais

Guilherme Greco
LinkedIn: @guilhermesgreco

Thiago Morais
LinkedIn: @thiagooliveira