AWS re:Invent

DAT335-R1

Build serverless applications with Amazon DynamoDB and AWS Lambda

Jeff Demuth
Solutions Architect
Amazon Web Services

Sang Kong
Senior Solutions Architect
Amazon Web Services

AWS re:Invent

aws

Building serverless applications with Amazon DynamoDB, AWS Lambda & more: Workshop

1. Required: **Laptop** and **power supply**
2. Go to **https://dashboard.eventengine.run**
3. Type in your hash

Agenda

Service overviews + hands-on labs

Advanced concepts

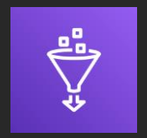
Prerequisites: Basic knowledge of DynamoDB, Lambda, Amazon Simple Storage Service (Amazon S3), and Amazon Kinesis


Amazon
API Gateway


Amazon
Athena


Amazon
Cognito


Amazon
DynamoDB


AWS Glue


Kinesis


Lambda


Amazon
Simple Storage Service

Scenario

More zombies!

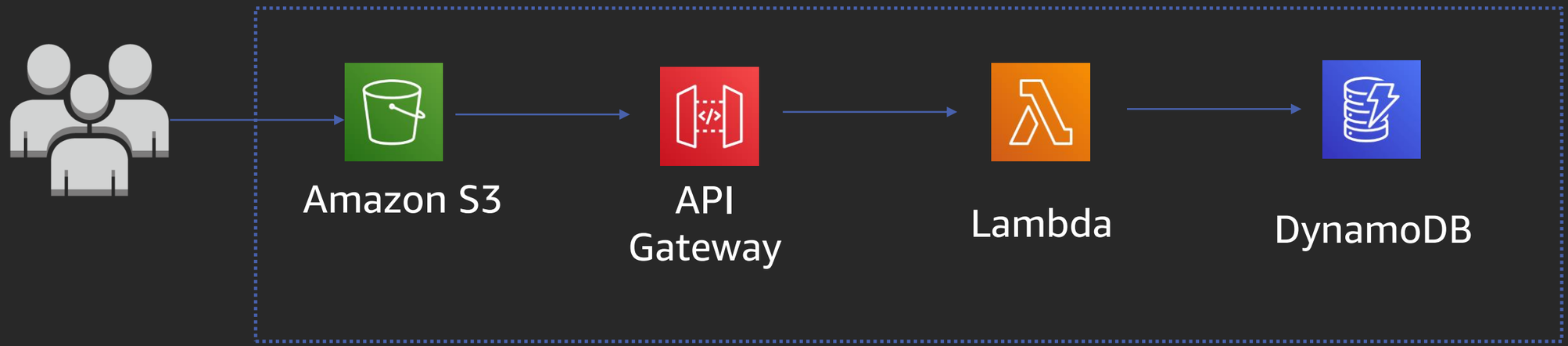
Why serverless?

- Pay by usage
- No server management
- Flexible scaling
- Automated high availability
- Ease of getting started

“Build and run applications without thinking about servers”

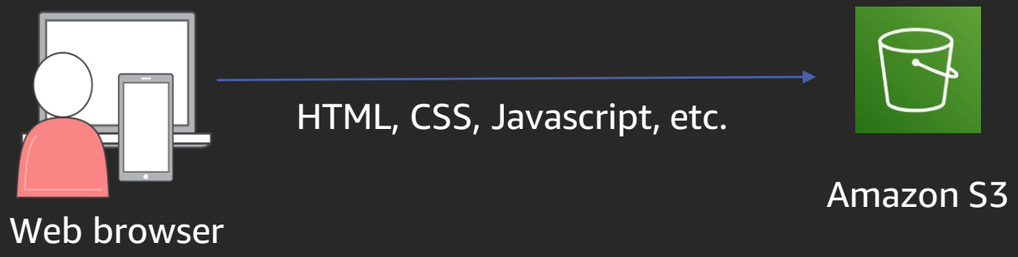
Solution Overview (Enterprise)

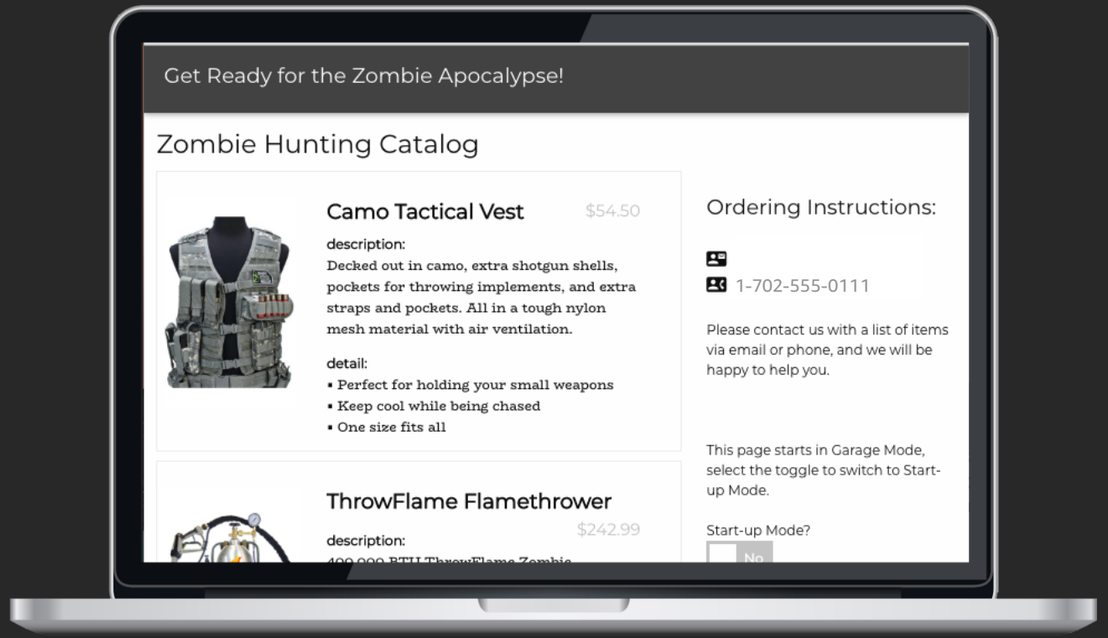
- Amazon S3
- API Gateway
- Lambda
- DynamoDB



Amazon S3

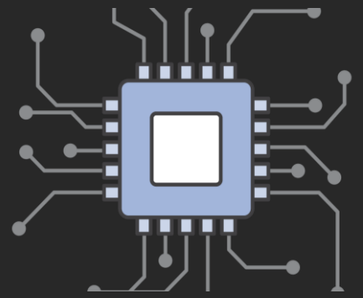
Use Amazon S3 to host serverless websites





- Files need to be public. Bucket does not.
- Distribute with Amazon CloudFront
- Make dynamic with Lambda@Edge
- Protect site with AWS Shield Advanced and AWS WAF

API Gateway



Create a unified API frontend for multiple microservices

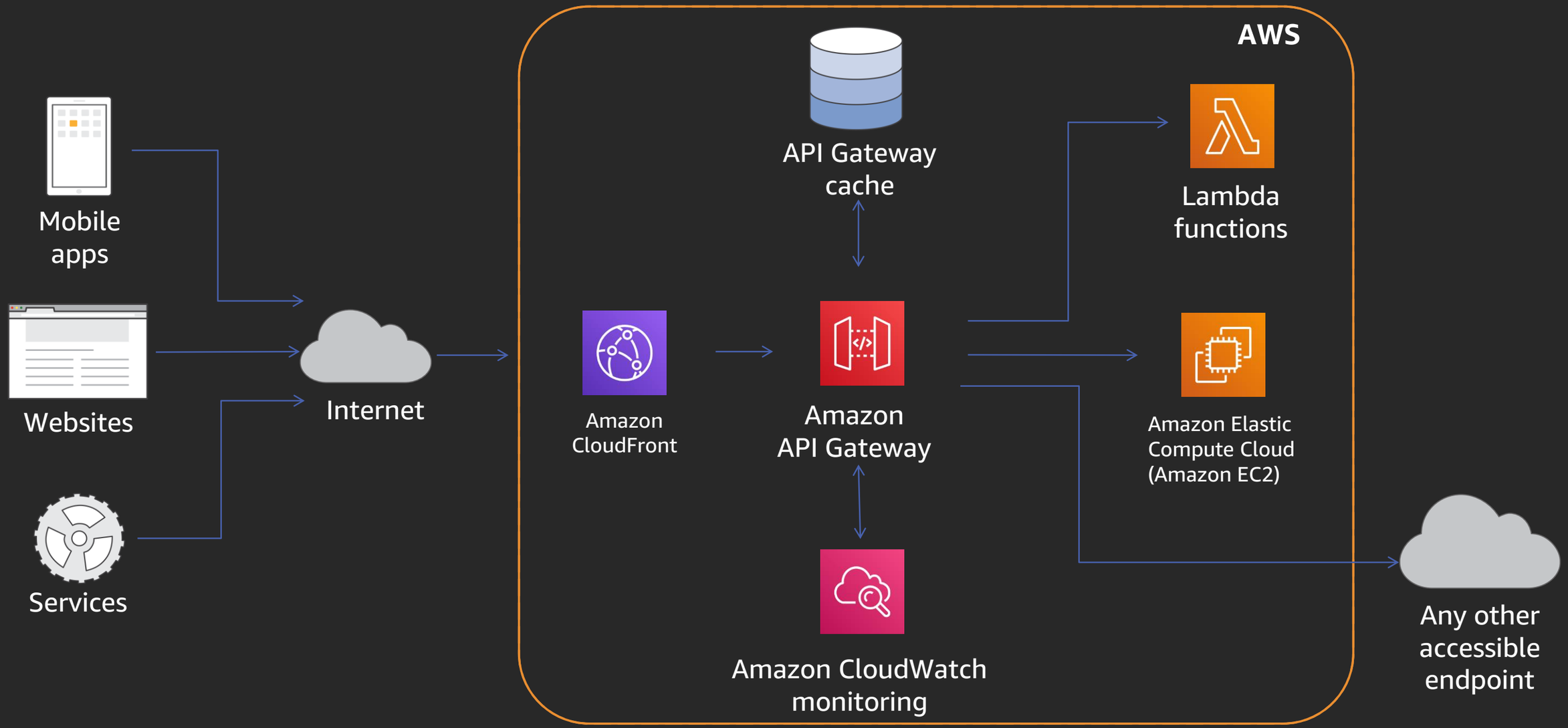
DDoS protection and throttling for your backend

Authenticate and authorize requests to a backend

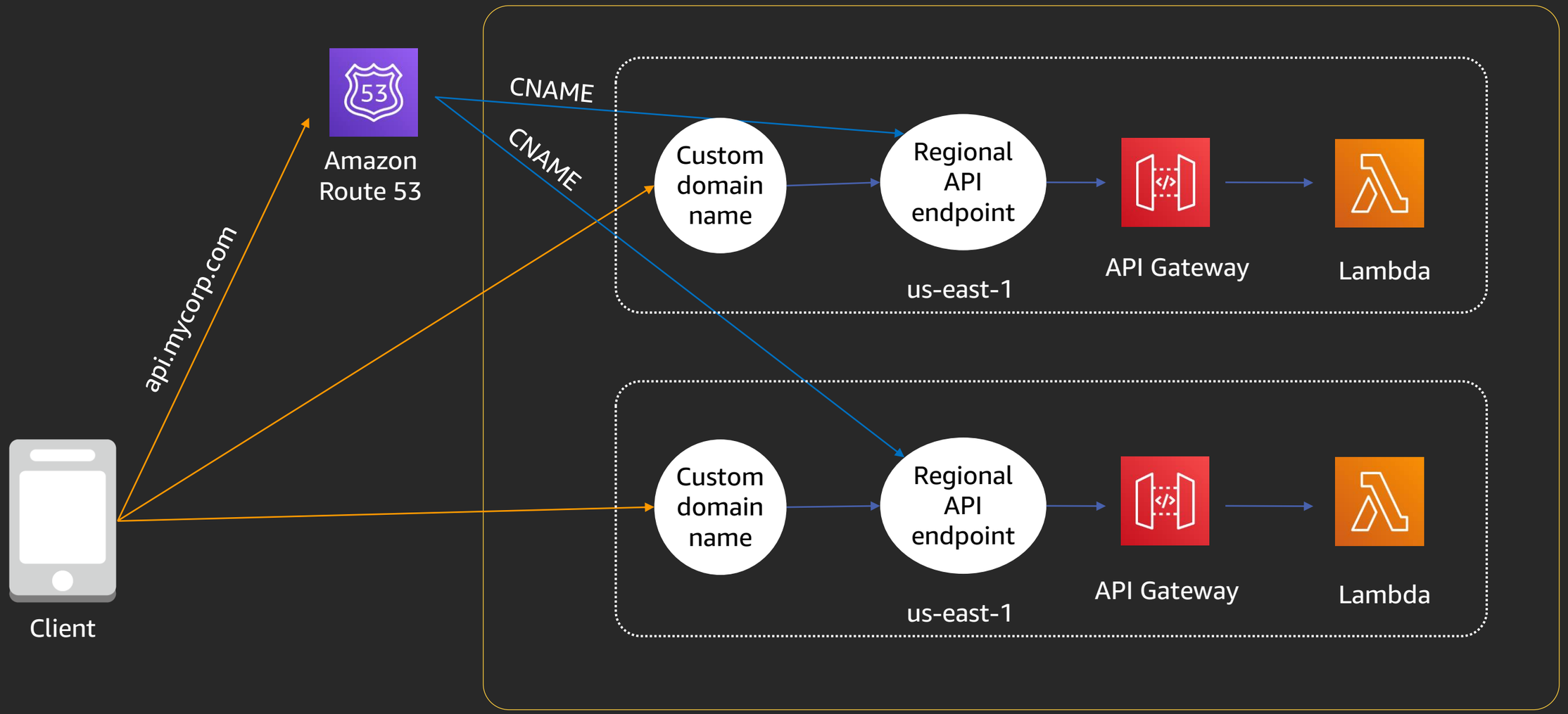
Throttle, meter, and monetize API usage by third-party developers

“Create, maintain, and secure APIs at any scale”

API Gateway – API call flow

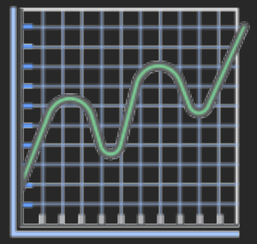


API Gateway – Multi-region example



Lambda

No servers to manage
Focus on code



Continuous scaling
Configure concurrency

Sub second metering
Pay for only what you use

Compute service that lets you run code without managing servers: Scale, monitor, and, trigger on your behalf

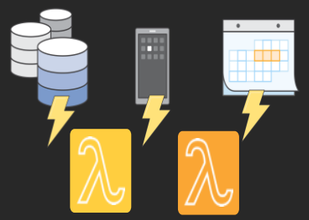
Lambda

Bring your own code

- Node.js, Java, Python, C#, Go, Ruby, etc.

Simple resource model

- Select power rating from 128 MB to 3 GB
- CPU and network allocated proportionately



Flexible use

- Synchronous or asynchronous
- Integrated with other AWS services

Stateless

- Persist data using external storage
- No affinity or access to underlying infrastructure

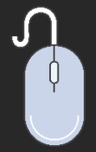
Lambda – Good practices

- Minimize package size to only what is necessary
- Separate the Lambda handler from core logic
    - Leverage container reuse
- Use environment variables to modify operational behavior
- Self-contain dependencies in your function package
- Leverage “max memory used” to right-size your functions
- Understand, set, and monitor concurrency
- Delete large unused functions (75GB limit)
- Leverage ecosystem—AWS Serverless Application Model (AWS SAM), AWS CodePipeline, AWS CodeDeploy, AWS X-Ray, AWS Cloud9

Lambda – **fnCreateOrder** CODE

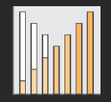
```
ddbclient = boto3.client('dynamodb')
def lambda_handler(event, context):
    data = json.loads(event['body'])
    shipping_option = data['shipping_option']
    for liit in data['line_items']:
        product = liit['product']
        qty = str(liit['qty'])
        price = str(liit['price'])
        ddbclient.put_item(
    TableName='UnifiedTable',
    Item={'pk':{'S': order_id},
        'sk':{'S': product},
        'ordercreated':{'S':str(dt)},
        'qty':{'N':qty},
        'price':{'S':price},
        'shipping_option':{'S':shipping_option}}})
return {'statusCode': '200', 'body': json.dumps(resp),
        'headers': {'Content-Type': 'application/json','Access-Control-Allow-Origin': '*‘ }}
```

DynamoDB



Fully managed NoSQL

Document or key-value

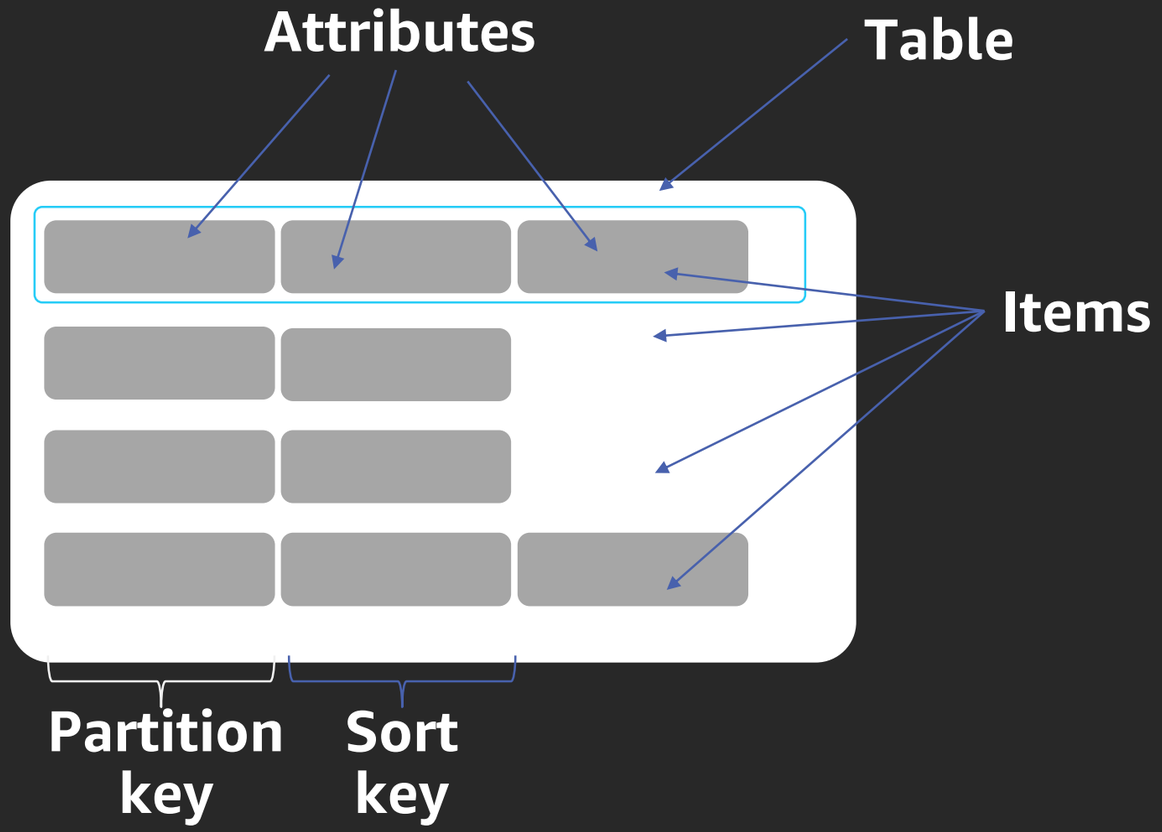


Scales to any workload

Fast and consistent

Access control

Event-driven programming



- **Local secondary index**
- **Global secondary index**

Amazon DynamoDB is a nonrelational database that delivers reliable performance at any scale.

DynamoDB – Tenets

- Understand the use case
- Identify the access patterns
    - Read/write workloads
    - Query dimensions and aggregations
- Data-modeling
    - Using NoSQL design patterns
    - Amazon DynamoDB Streams, single table design, sparse index, overloading, etc.
    - https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/best-practices.html

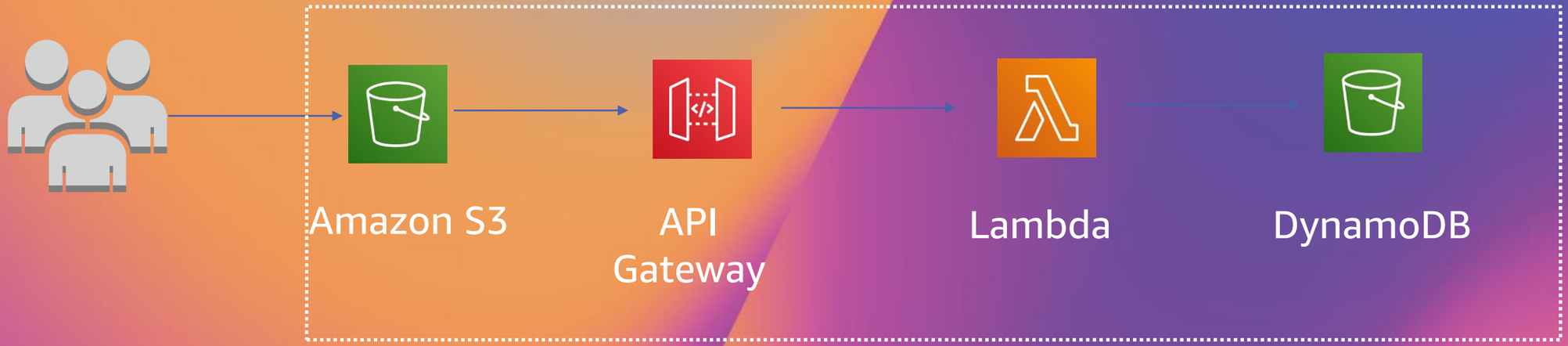
DynamoDB – Single Table Design

Global Secondary Index (GSI) pk sk

Sparse Index

GSI Overloading

pk	sk	itemType	qty	blnOrdered	description	shipping_option	status	ordered
CUST-ORDER-153213426	Camo Tactical Vest		7			Drone Delivery	Created	
CUST-ORDER-153213531	Super Quiet Pisto…		4			1-day shipping	Created	
VEND-ORDER-2018-7-21	Camo Tactical Vest		200					
totalorders	Metrics	metrics	20					
Zombie Juicery Energy …	Zombie Juicery E…	part	194	0	Zombie Juicery En…			6
Super Quiet Pistol Cross…	Super Quiet Pisto…	part	196	0	Super Quiet Pistol…			4
Slayer Double Headed Axe	Slayer Double He…	part	200	0	Slayer Double Hea…			0
Black Undead Water Bot…	Black Undead Wa…	part	200	0	Black Undead Wat…			0
ThrowFlame Flamethro…	Throwflame Fla…	part	199	0	ThrowFlame Flam…			1



Let’s build it!

Follow instructions listed under **“Phase 1 - Small Enterprise”**

Serverless zombie shopping

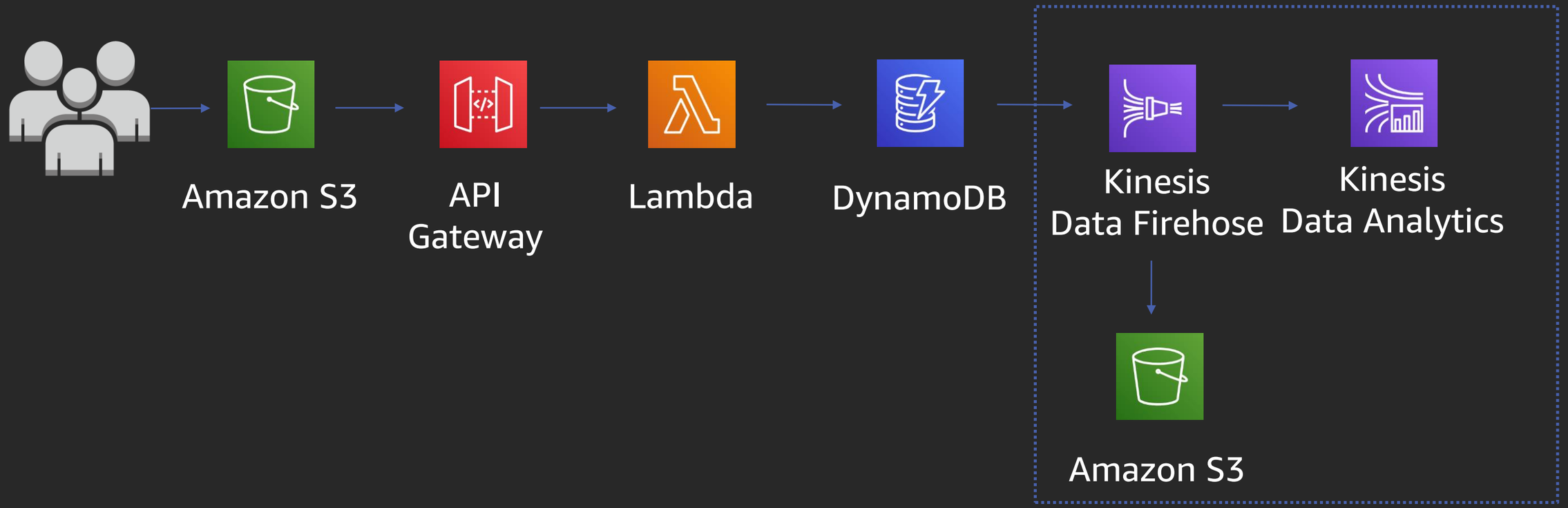
+ Analytics

AWS re:Invent

aws

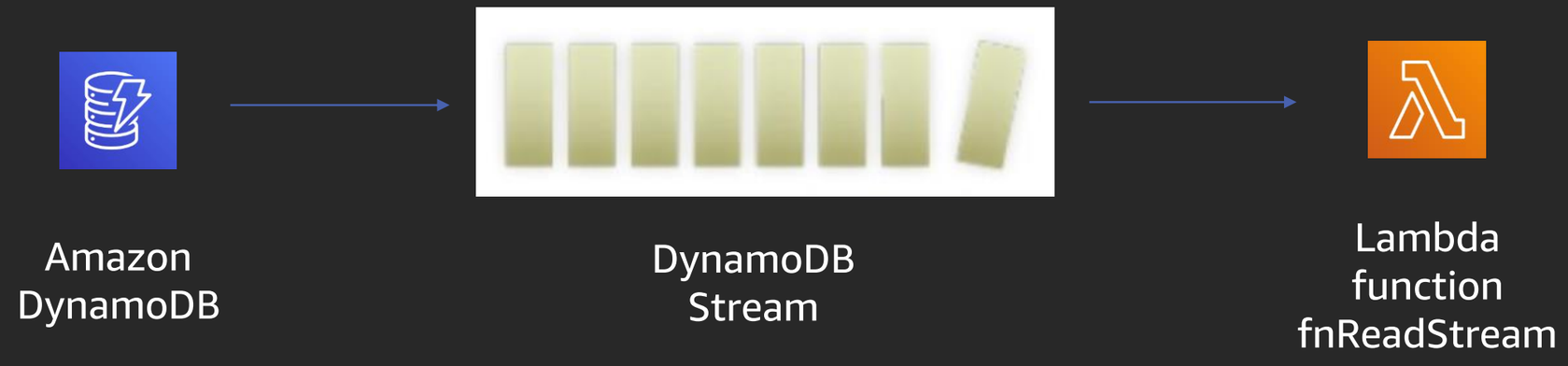
Solution overview (analytics)

- Kinesis Data Firehose
- Kinesis Data Analytics

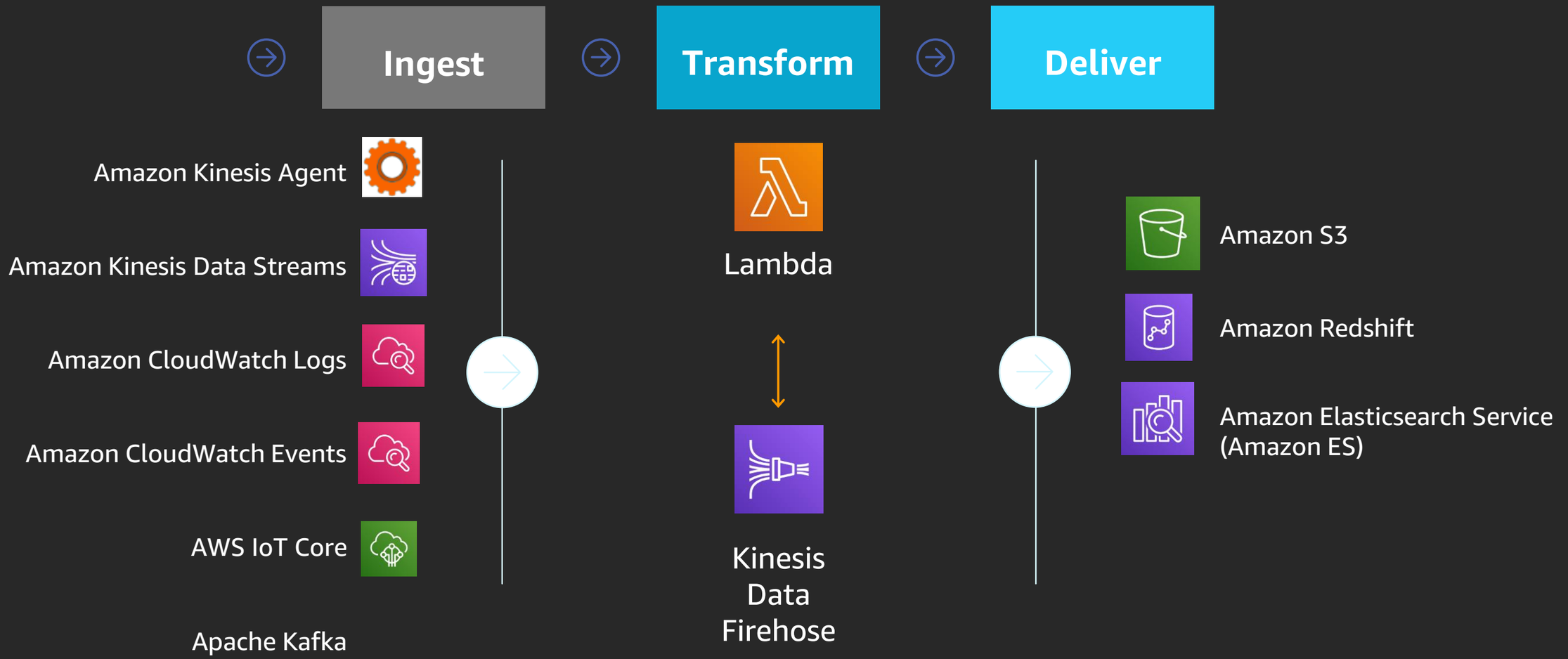


DynamoDB Streams

- DynamoDB Streams captures a time-ordered sequence of item-level modifications (stored up to 24 hours) in a DynamoDB table
- Process multiple modifications at a time with Lambda



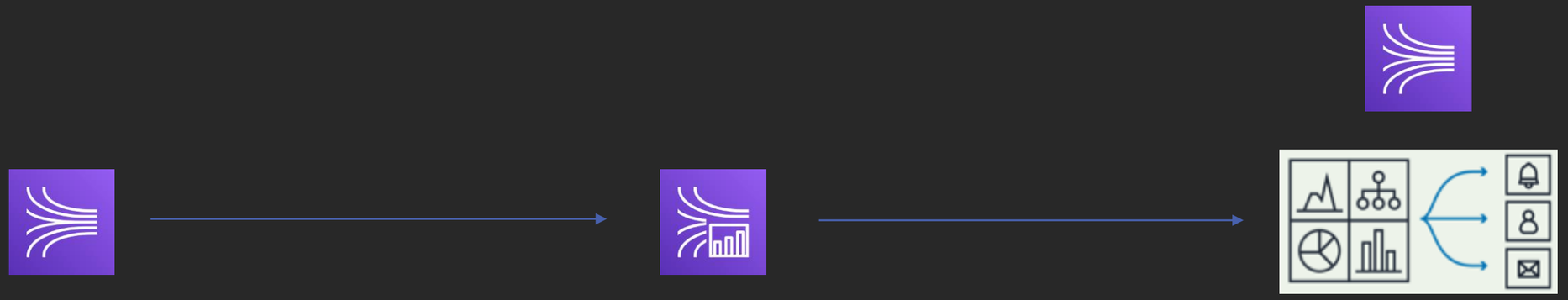
Kinesis Data Firehose



"Amazon Kinesis Data Firehose is the easiest way to reliably load streaming data into data lakes, data stores and analytics tools."

https://aws.amazon.com/kinesis/data-firehose/?nc=sn&loc=2&dn=3

Kinesis Data Analytics



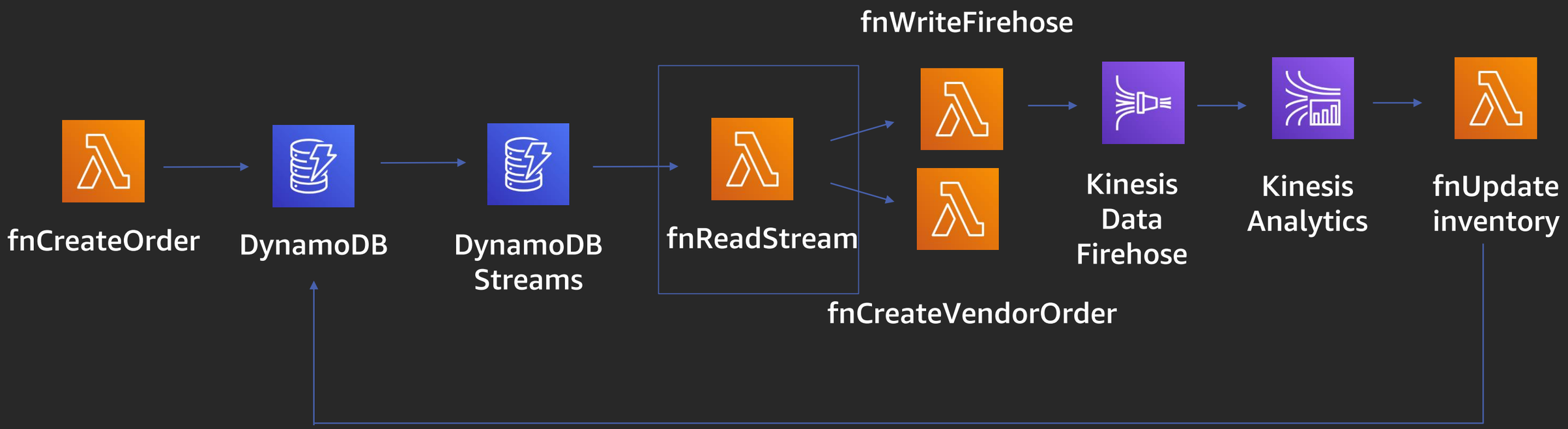
Input
Capture streaming data with Kinesis Data Firehose or Kinesis Data Streams

Kinesis Data Analytics
Run standard SQL queries against data streams

Output
Send processed data to analytics tools so you can create alerts and respond in real time, or Kinesis, or Lambda

Amazon Kinesis Data Analytics is the easiest way to process streaming data in real time with standard SQL without having to learn new programming languages or processing frameworks.

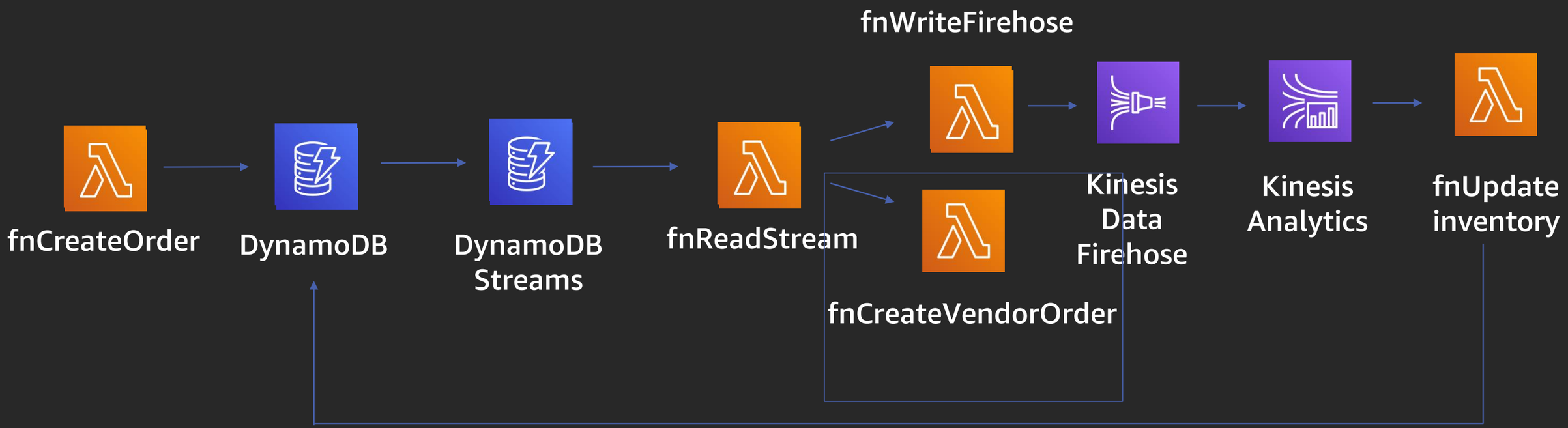
Analytics Solution – All Lambda functions



Lambda – **fnReadStream** CODE

```
import boto3
import os
write_firehose = 'fnWriteFirehose'
create_vendor_order = 'fnCreateVendorOrder'
lclient = boto3.client('lambda')
def lambda_handler(event, context):
    logger = logging.getLogger()
    logger.setLevel(logging.INFO)
    try:
        payload = event['Records']
        lclient.invoke(FunctionName=write_firehose, InvocationType='Event', Payload=json.dumps(event))
        lclient.invoke(FunctionName=create_vendor_order, InvocationType='Event', Payload=json.dumps(event))
    except Exception as ex:
        logging.info(str(ex))
```

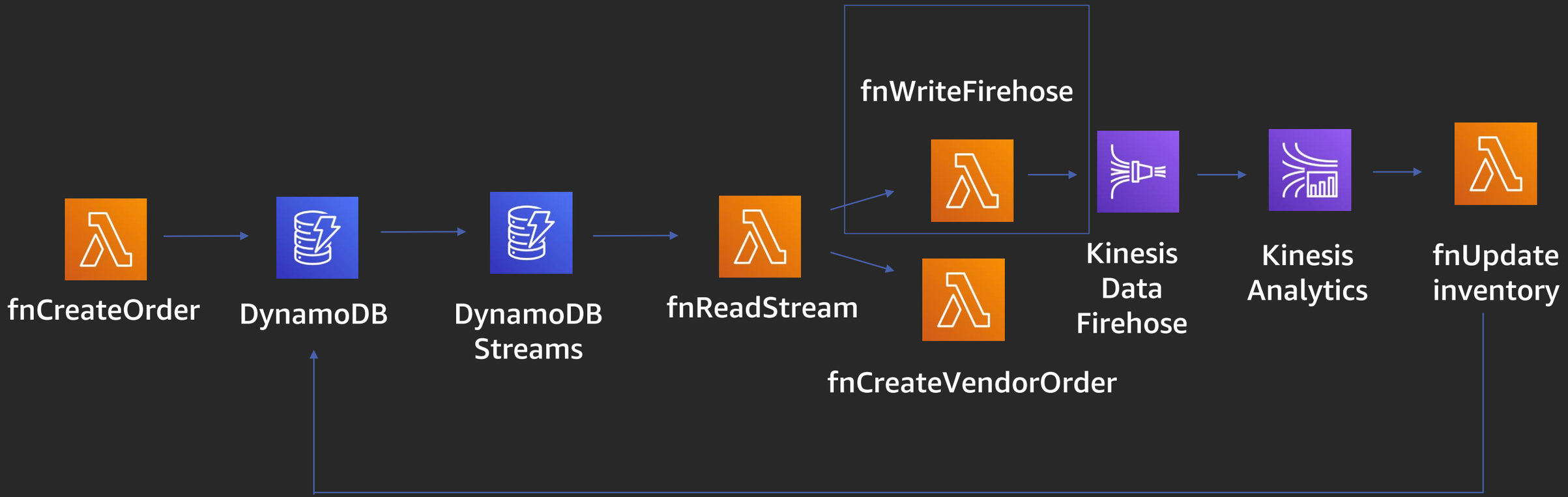
Analytics Solution – All Lambda functions



Lambda – fnCreateVendorOrder CODE

```
import boto3
ddbclient = boto3.client('dynamodb')
def lambda_handler(event, context):
      if int(qty) < 50:
            ddbclient.update_item(TableName='UnifiedTable', Key={'pk': {'S': product}, 'sk': {'S':
product}}, UpdateExpression='SET blnOrdered = :val', ExpressionAttributeValues={':val': {'N': '1'}})
```

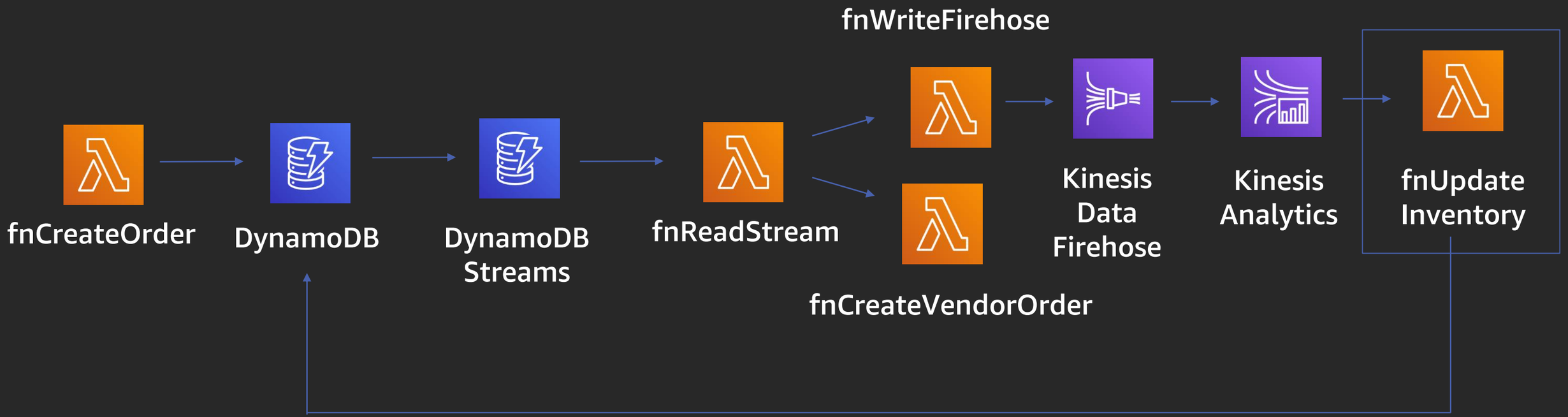
Analytics Solution – All Lambda functions



Lambda – fnWriteFirehose CODE

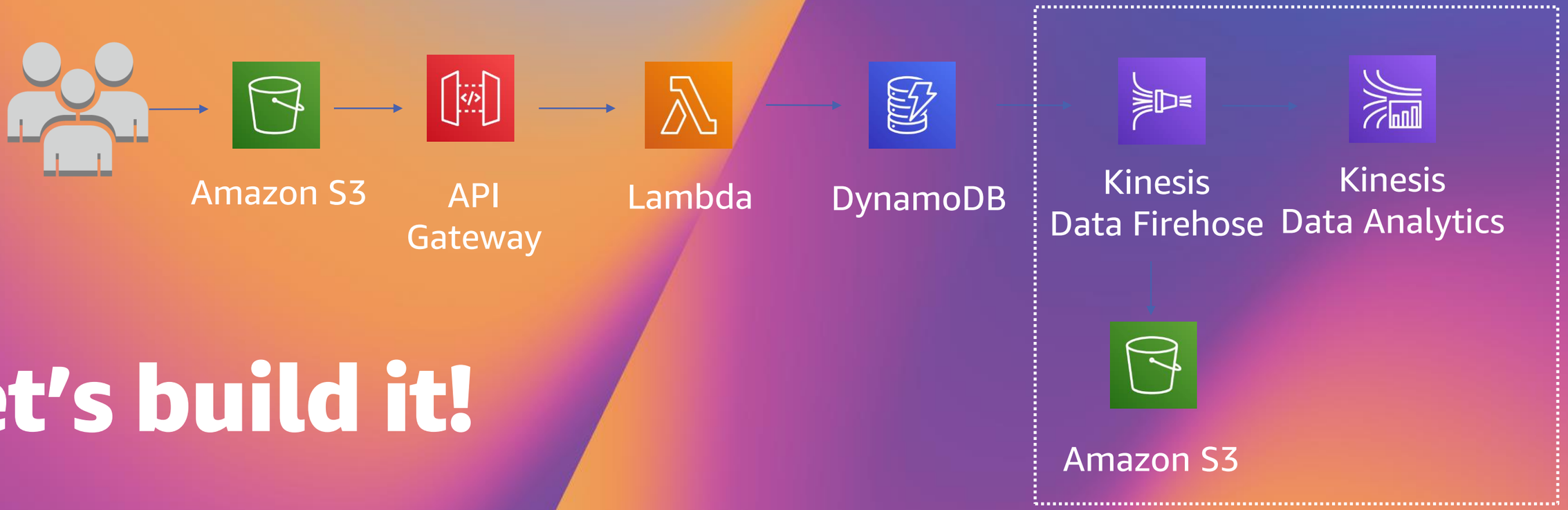
```
def lambda_handler(event, context):
        for item in event['Records']:
            dynamodb_dump = json.dumps(item['dynamodb'])
            new_image = item['dynamodb']['NewImage']
            output = _create_stream_data(new_image)
            print(kclient.put_record_batch(DeliveryStreamName=kinesis_stream_name, Records=records))
```

Analytics Solution – All Lambda functions



Lambda – fnUpdateInventory CODE

```
ddbclient = boto3.client('dynamodb')
def lambda_handler(event, context):
      for record in event.get('records', []):
        ddbclient.update_item(TableName='UnifiedTable', Key={'pk': {'S': product}, 'sk': {'S': product}},
UpdateExpression='SET ordered = ordered + :val', ExpressionAttributeValues={':val': {'N': str(quantity)}})
```



Let’s build it!

Analytics (30 minutes)

1. Follow instructions listed under “Phase 2 – Large Enterprise”

Serverless zombie shopping

+ Analytics

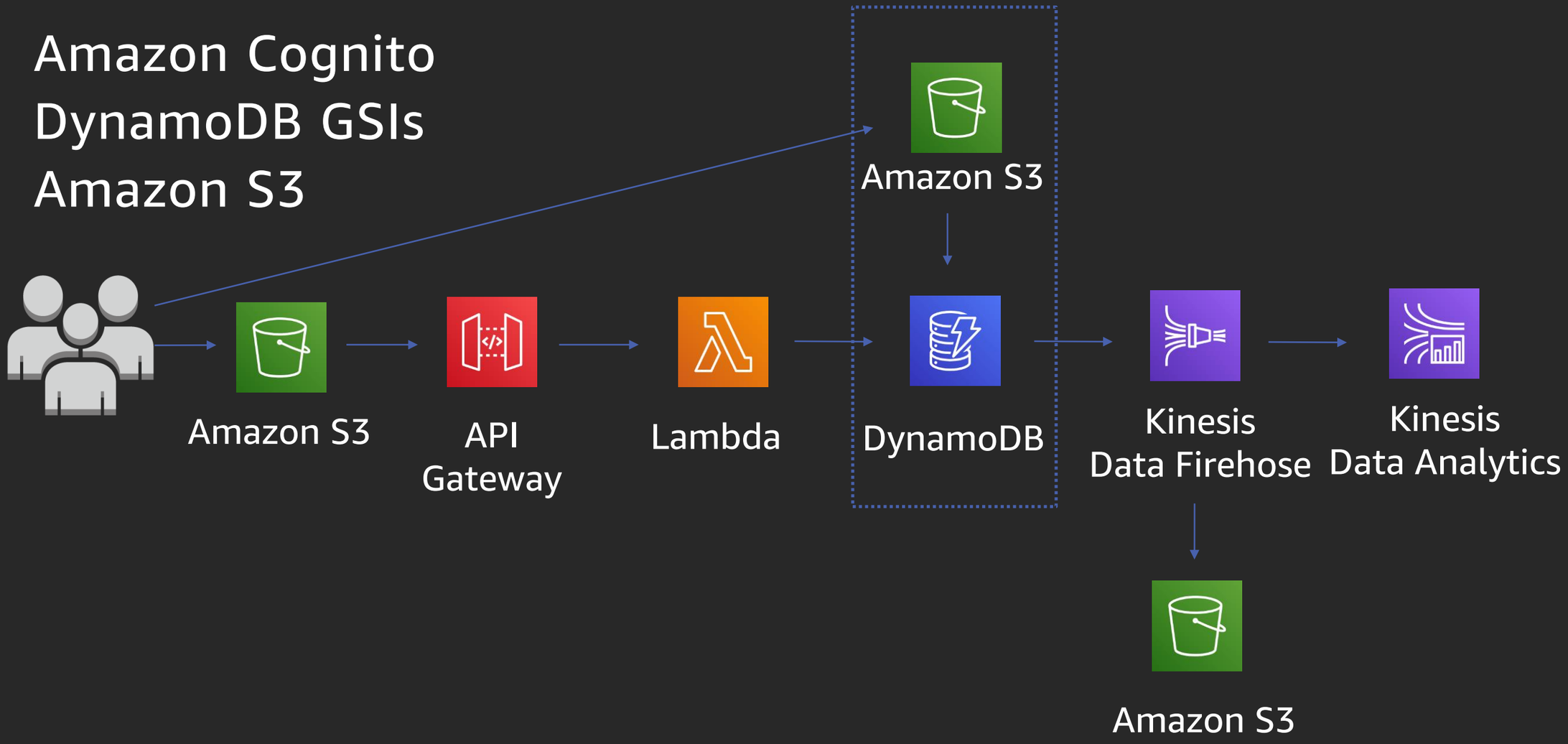
+ Operational Metrics

AWS re:Invent

aws

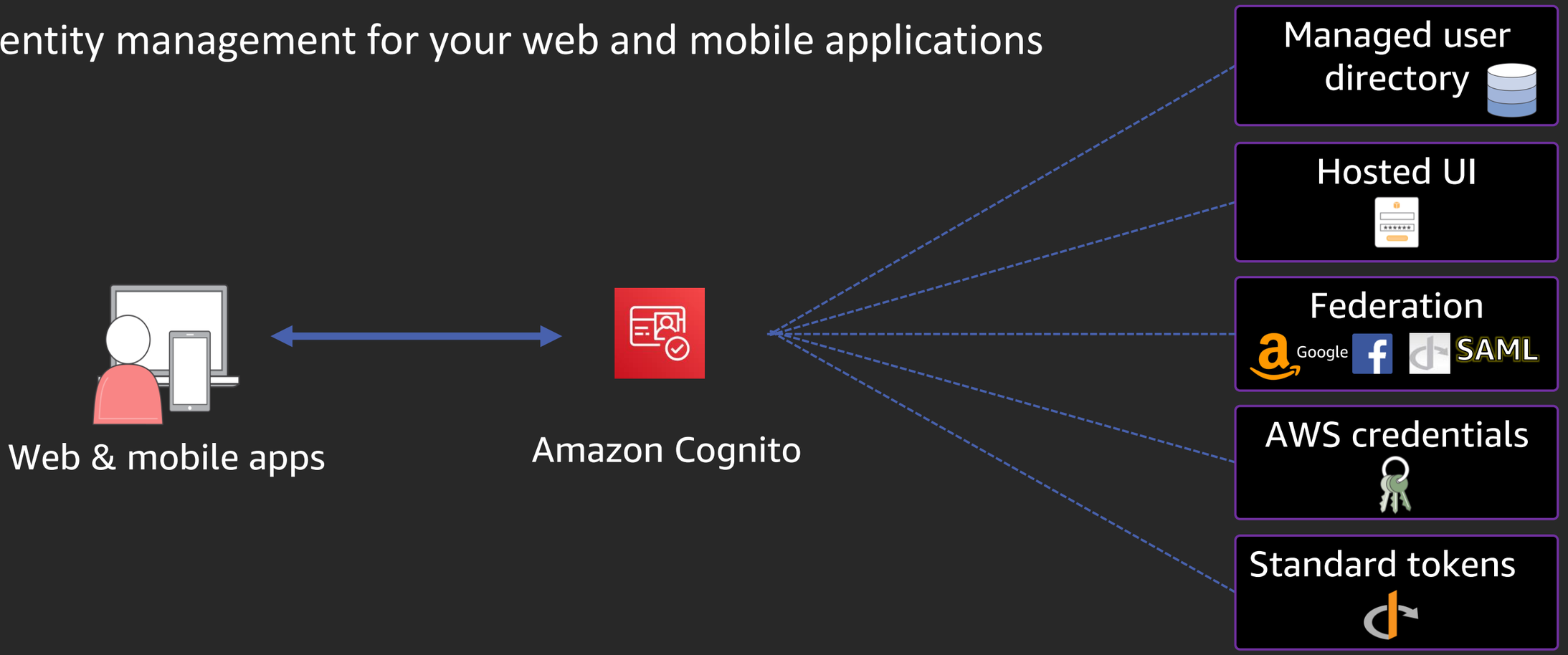
Solution overview (operational metrics)

- Amazon Cognito
- DynamoDB GSIs
- Amazon S3



Amazon Cognito

Identity management for your web and mobile applications



Amazon Cognito simplifies the task of authenticating users and storing, managing, and syncing their data across multiple devices, platforms, and applications.

Amazon Cognito

User pools and identity federation

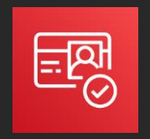
User pools

Managed user directory

Hosted UI

Federated UI

Google SAML



Amazon Cognito

Identity federation

AWS credentials

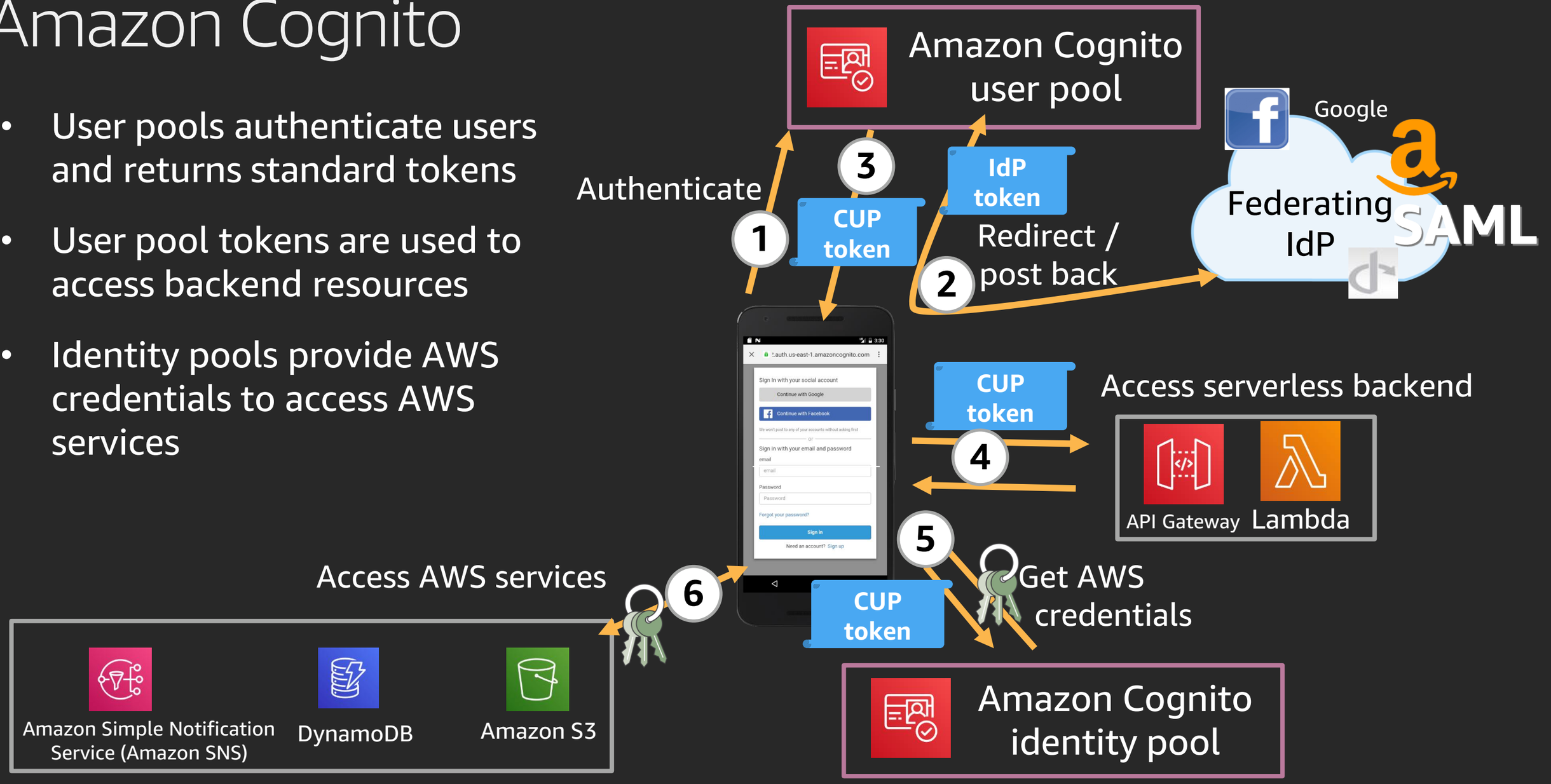
Standard tokens

Federated identity

Google SAML

Amazon Cognito

- User pools authenticate users and returns standard tokens
- User pool tokens are used to access backend resources
- Identity pools provide AWS credentials to access AWS services



Amazon Cognito

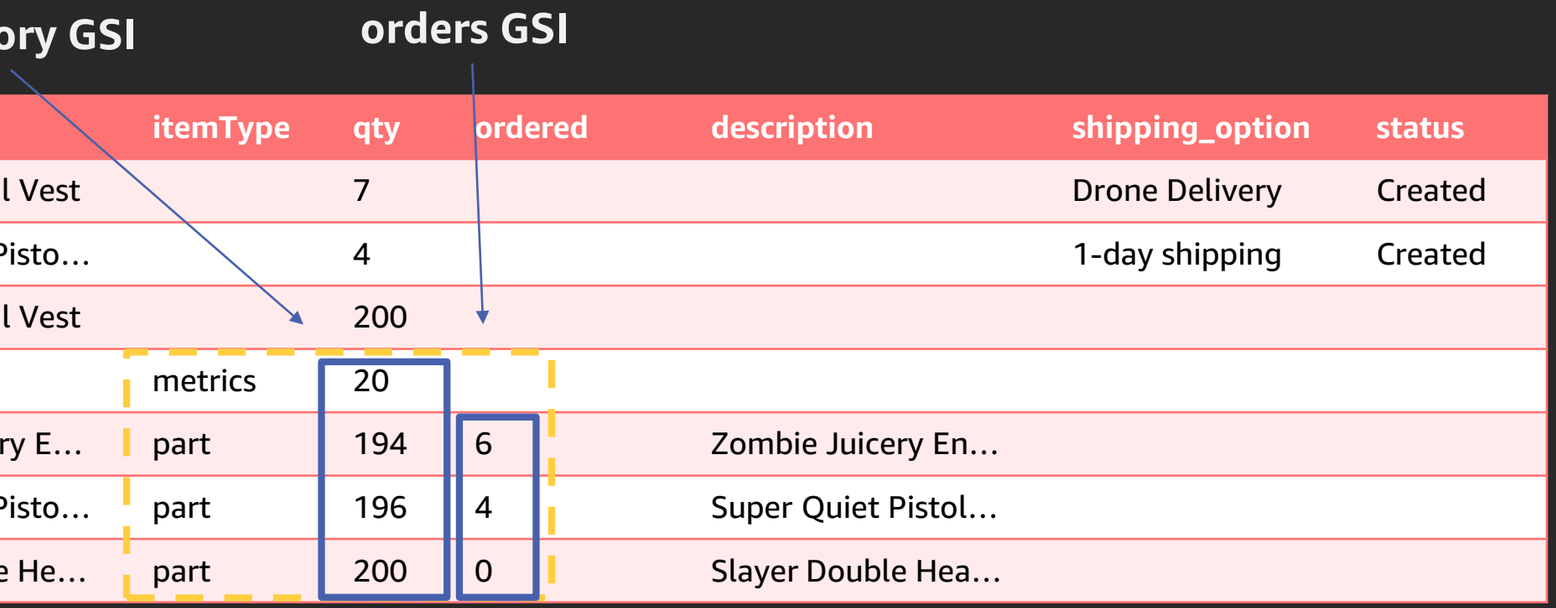
Additional Capabilities

- Integrated with API Gateway, AWS ALB, Amazon Pinpoint
- Advanced security
- Custom user flows with Lambda hooks
- AWS Amplify—declarative Javascript library for cloud development
- AWS AppSync—build data-driven apps with real-time and offline capabilities (GraphQL)

Unauthenticated flow

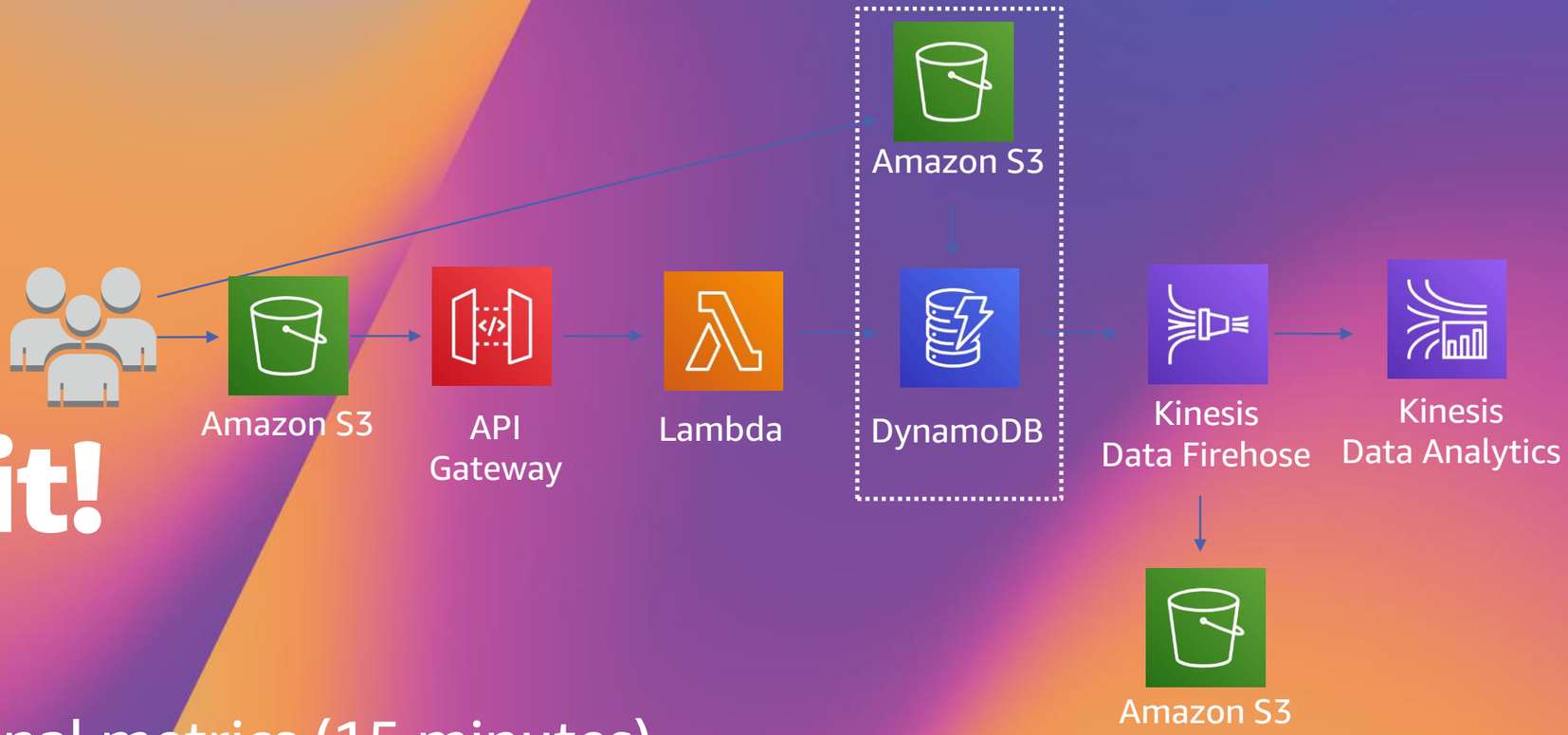
DynamoDB – Global secondary index (GSI)

- Global secondary index
- Index with a partition key and sort key that can be different from those on the base table
- Global because queries on the index can span all data in the base table, across all partition keys



pk	sk	itemType	qty	ordered	description	shipping_option	status
CUST-ORDER-153213426	Camo Tactical Vest		7			Drone Delivery	Created
CUST-ORDER-153213531	Super Quiet Pisto…		4			1-day shipping	Created
VEND-ORDER-2018-7-21	Camo Tactical Vest		200				
totalorders	Metrics	metrics	20				
Zombie Juicery Energy …	Zombie Juicery E…	part	194	6	Zombie Juicery En…		
Super Quiet Pistol Cross…	Super Quiet Pisto…	part	196	4	Super Quiet Pistol…		
Slayer Double Headed Axe	Slayer Double He…	part	200	0	Slayer Double Hea…		

Let’s build it!



Operational metrics (15 minutes)

1. Follow instructions listed under “Phase 3 – Operational Metrics”

Serverless zombie shopping

+ Analytics

+ Operational metrics

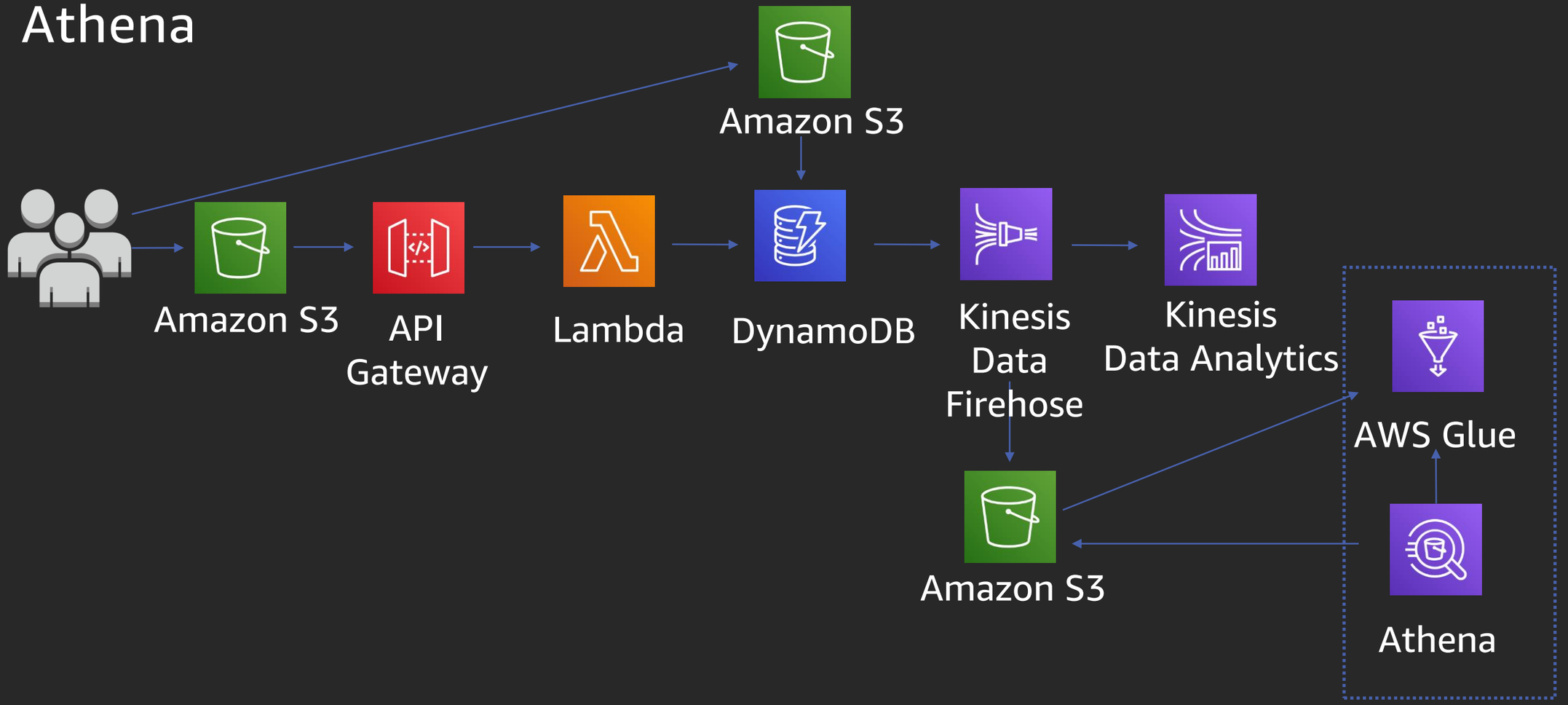
+ Business intelligence

AWS re:Invent

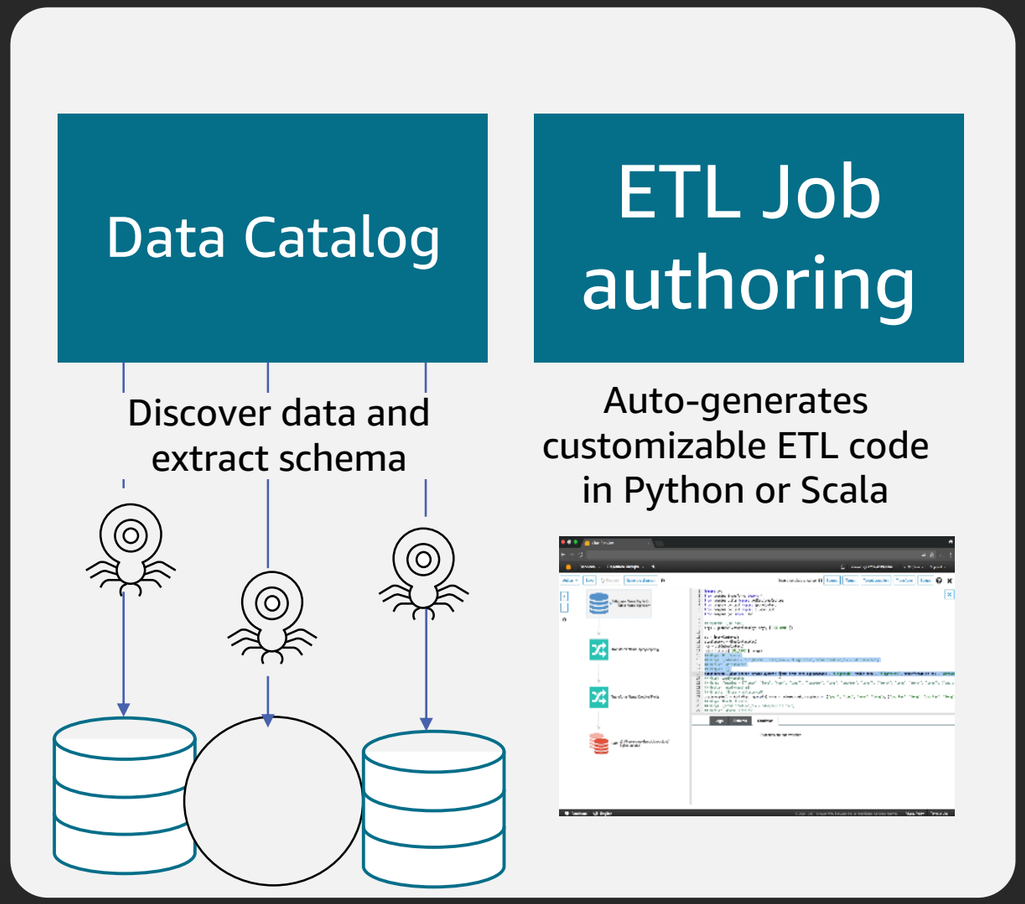
aws

Solution Overview (business intelligence)

- AWS Glue
- Athena



AWS Glue



- Automatically discovers data and stores schema
- Data is immediately searchable, and available for extract, transform, and load (ETL)
- Automatically generates customizable code
- Schedules and runs your ETL job

"AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics."

https://aws.amazon.com/glue/

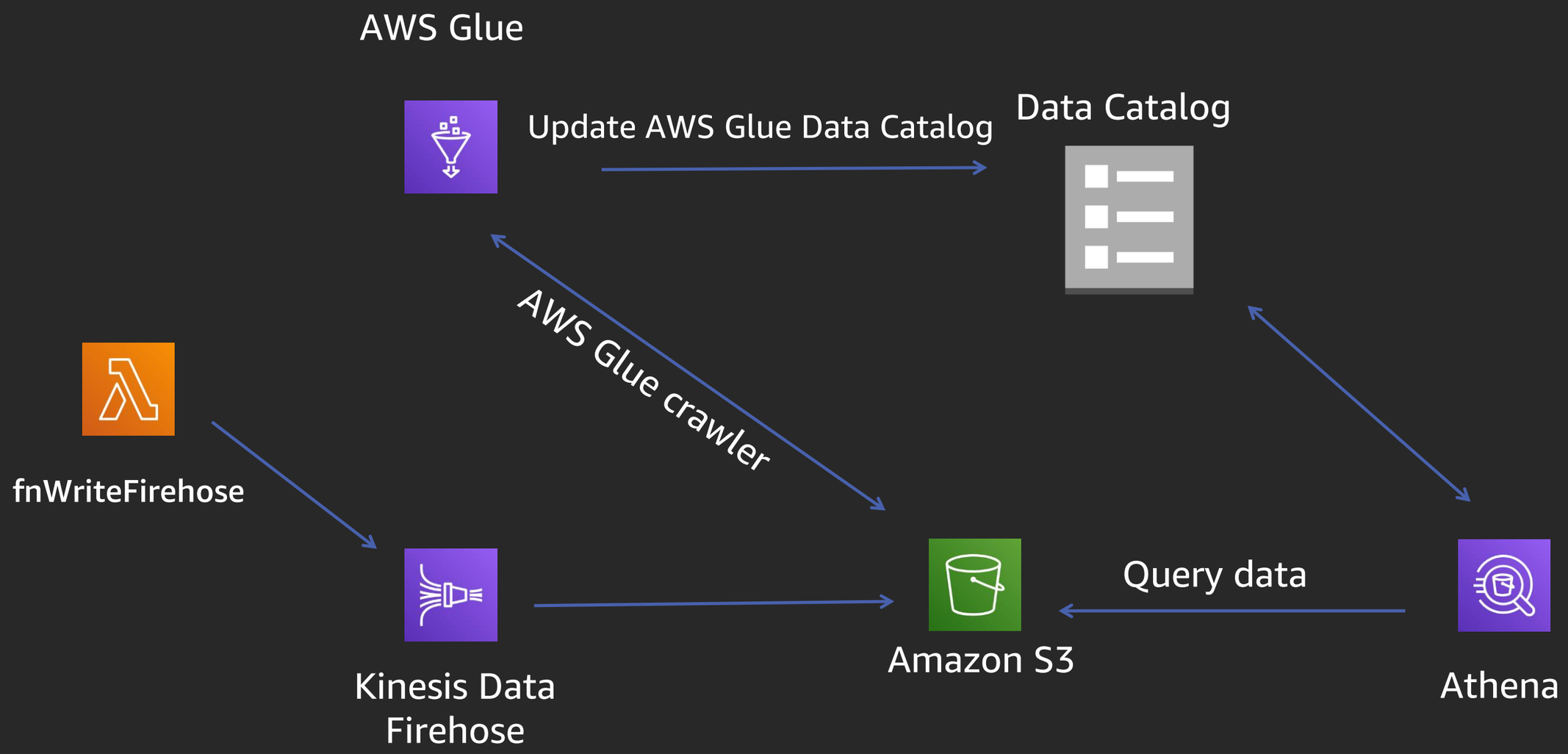
Athena

- Serverless—No infrastructure or administration, zero spin-up time, transparent upgrades
- Easy to use—Create a table (Hive DDL statement or add table wizard), query in ANSI SQL (complex joins, nested queries, etc.)
- Highly available—Uses warm compute pools across multiple Availability Zones
- Query Amazon S3 directly—No loading of data, query the raw format (text, csv, tsv, JSON, weblogs, AWS service logs, ORC, Parquet)

“Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.”

https://aws.amazon.com/athena/

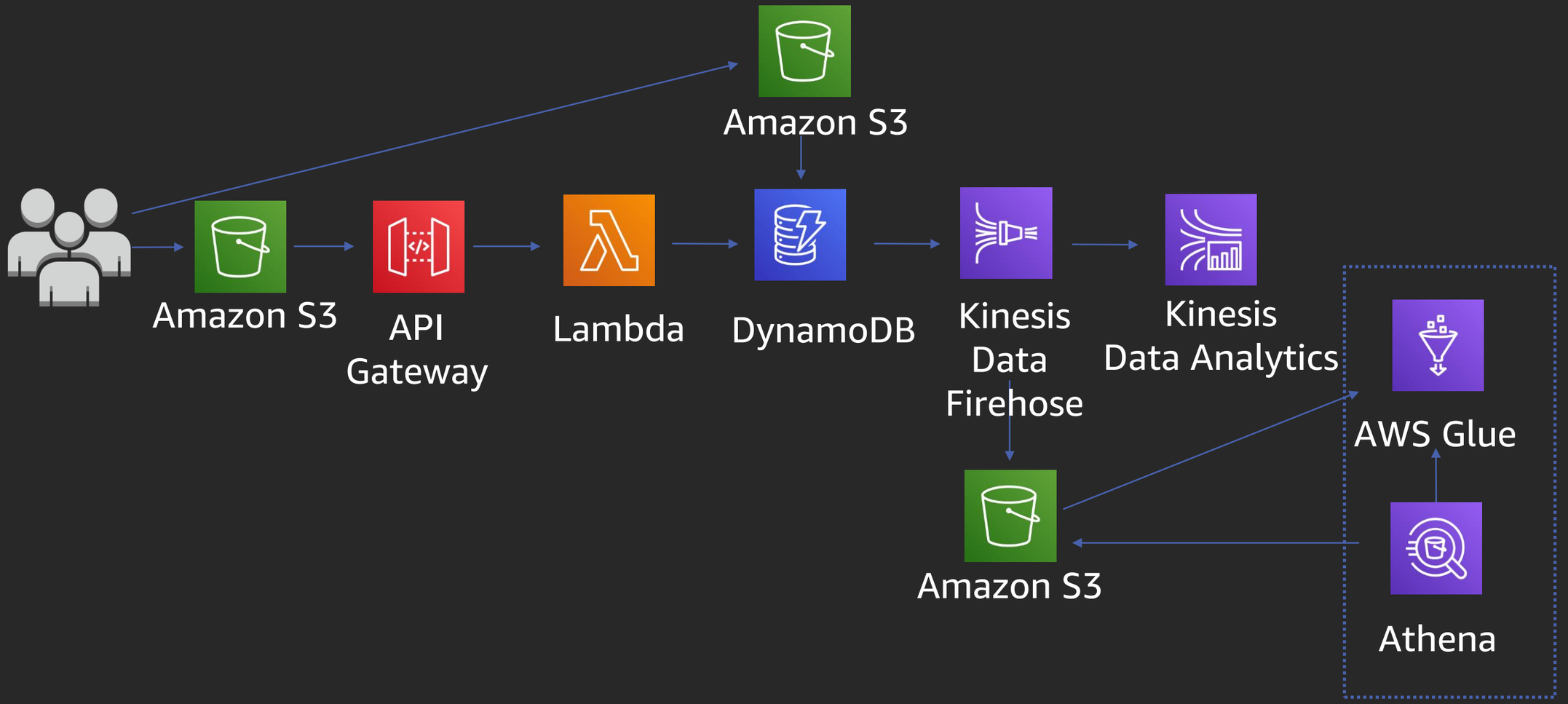
AWS Glue and Athena



Let’s build it!

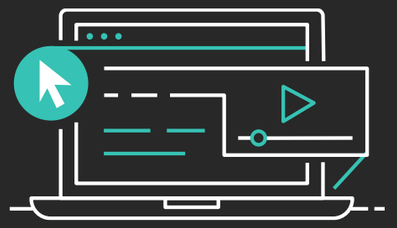
Business intelligence (15 minutes)

1. Follow instructions listed under Phase 4



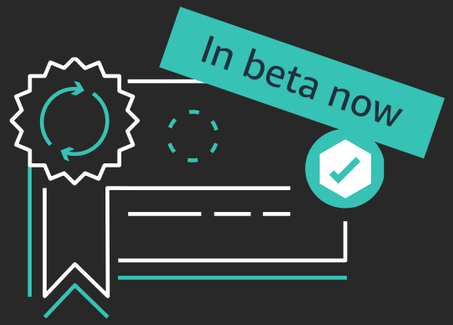
Learn databases with AWS Training and Certification

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



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

aws training and certification

Thank you!

AWS re:Invent

aws

Please complete the session survey in the mobile app.