
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SVS209

# Get started building your first serverless, event-driven application

Emily Shea

Head of Application Integration Go-To-Market
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Lots of great ideas…

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Lots of great ideas...

# ...limited time to build them

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# What you will know after this talk

- What AWS services to start with
- Best practices for developer tooling and testing
- How to evolve your application

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# What you will know after this talk

- What AWS services to start with
- Best practices for developer tooling and testing
- How to evolve your application

How serverless and event-driven architectures
let you **turn your ideas into applications faster**

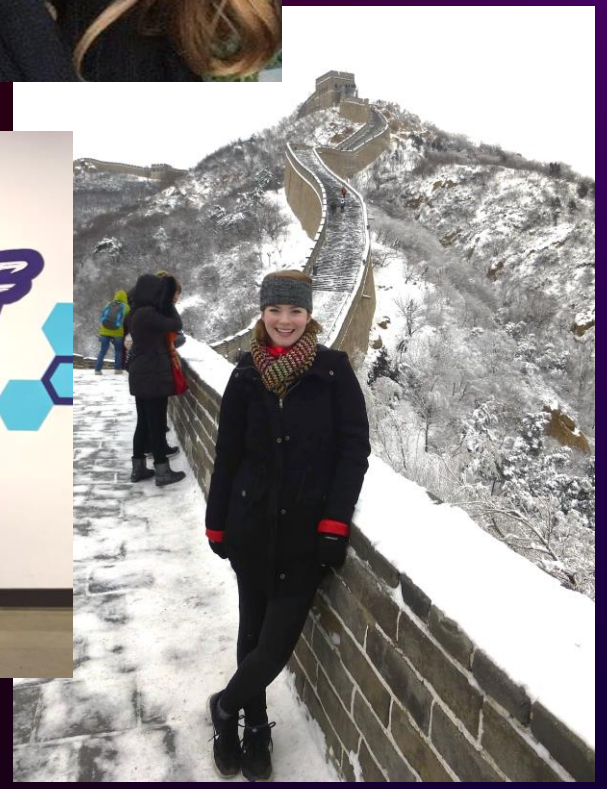
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# About me

- 6 years at Amazon
- Lead the Application Integration Go-to-Market team
- A self-taught serverless developer (Python, JavaScript)
- Studied the Chinese language





aws

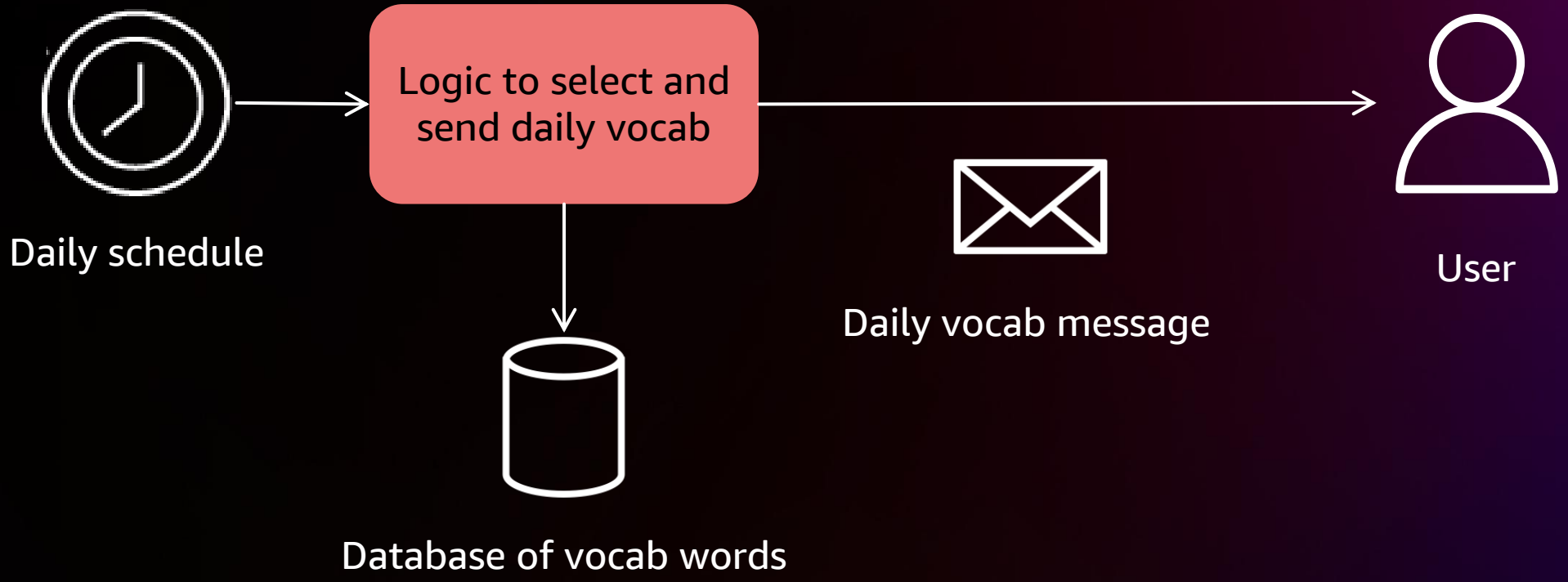
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Step 1: The app idea

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Language learning app

Daily reminders with Chinese vocabulary words




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# What other kinds of applications can you build?

An app that checks the weather and recommends what you should wear

An IoT camera app that identifies birds in your backyard

A home chore tracker with reminders and weekly leaderboards

An ecommerce application that scales to handle major sales automatically

A financial document processing pipeline that automates manual work

An integration platform that ingests Salesforce events to enrich customer order data

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Step 2: AWS serverless services

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Why serverless?

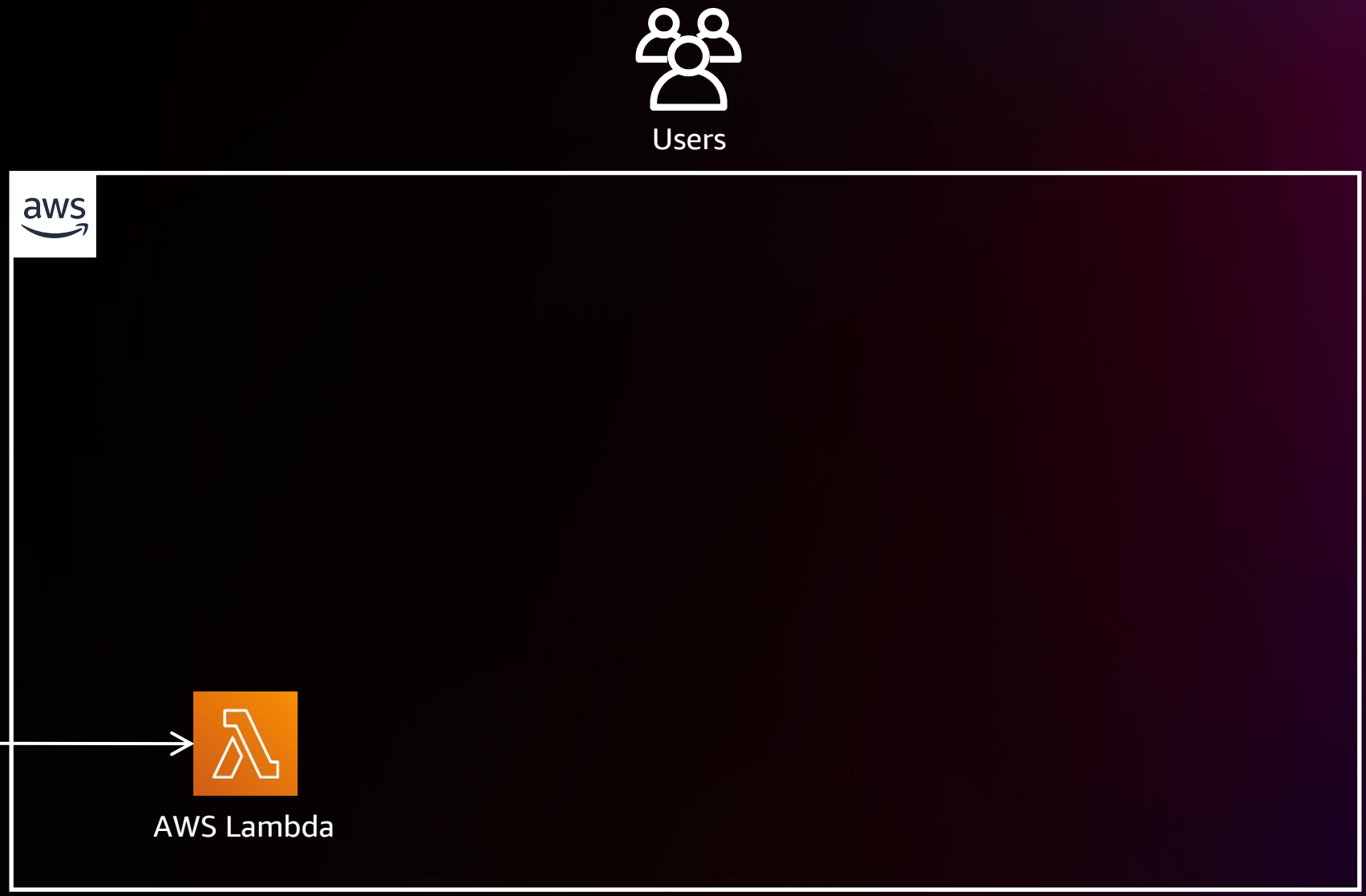
Serverless lets you **run code without managing servers**

- Scales up and down automatically
- Built-in high availability
- Pay-for-use billing model

Only code what you need to → Build your app faster

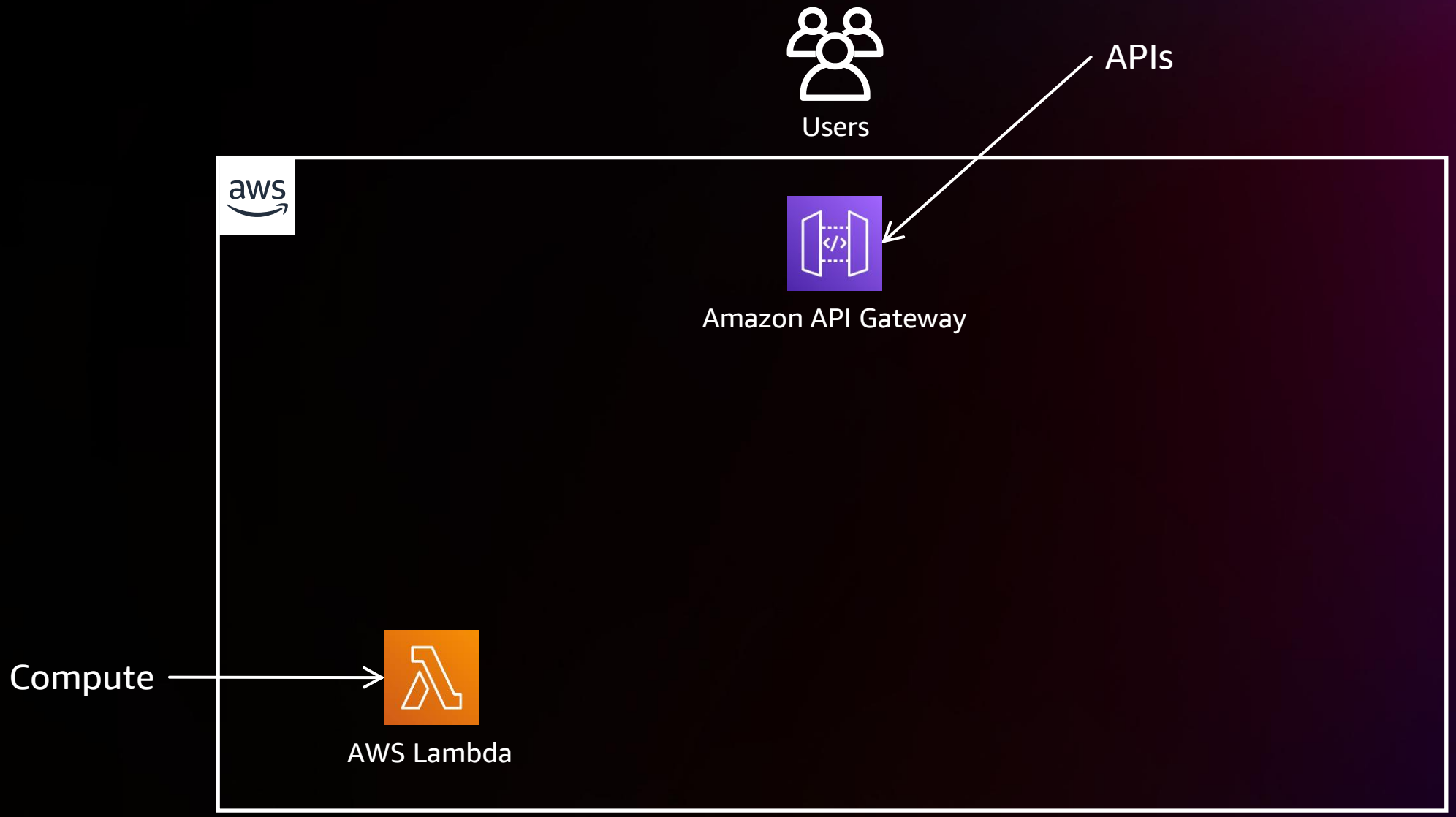
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The serverless toolbox



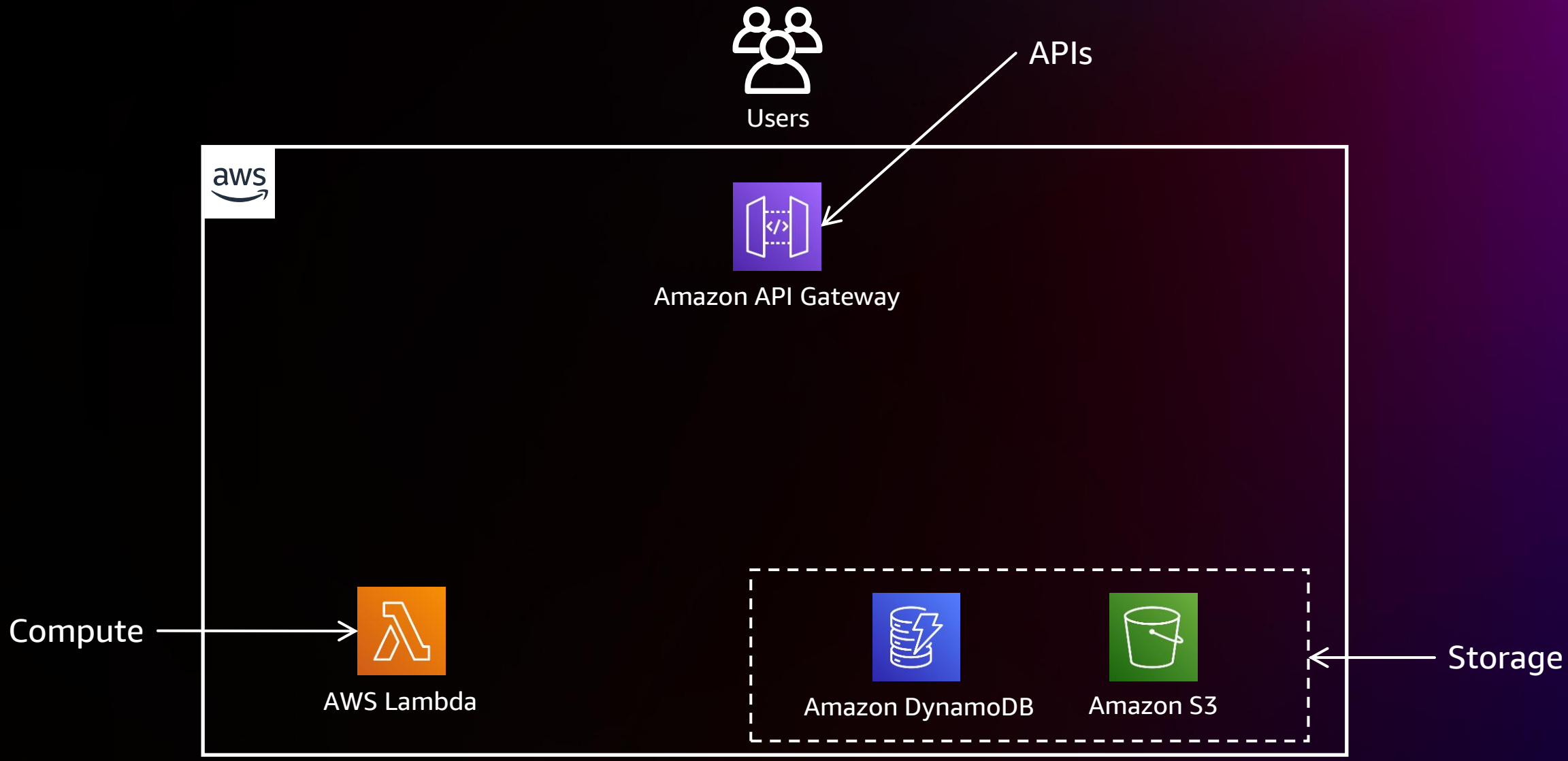

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The serverless toolbox



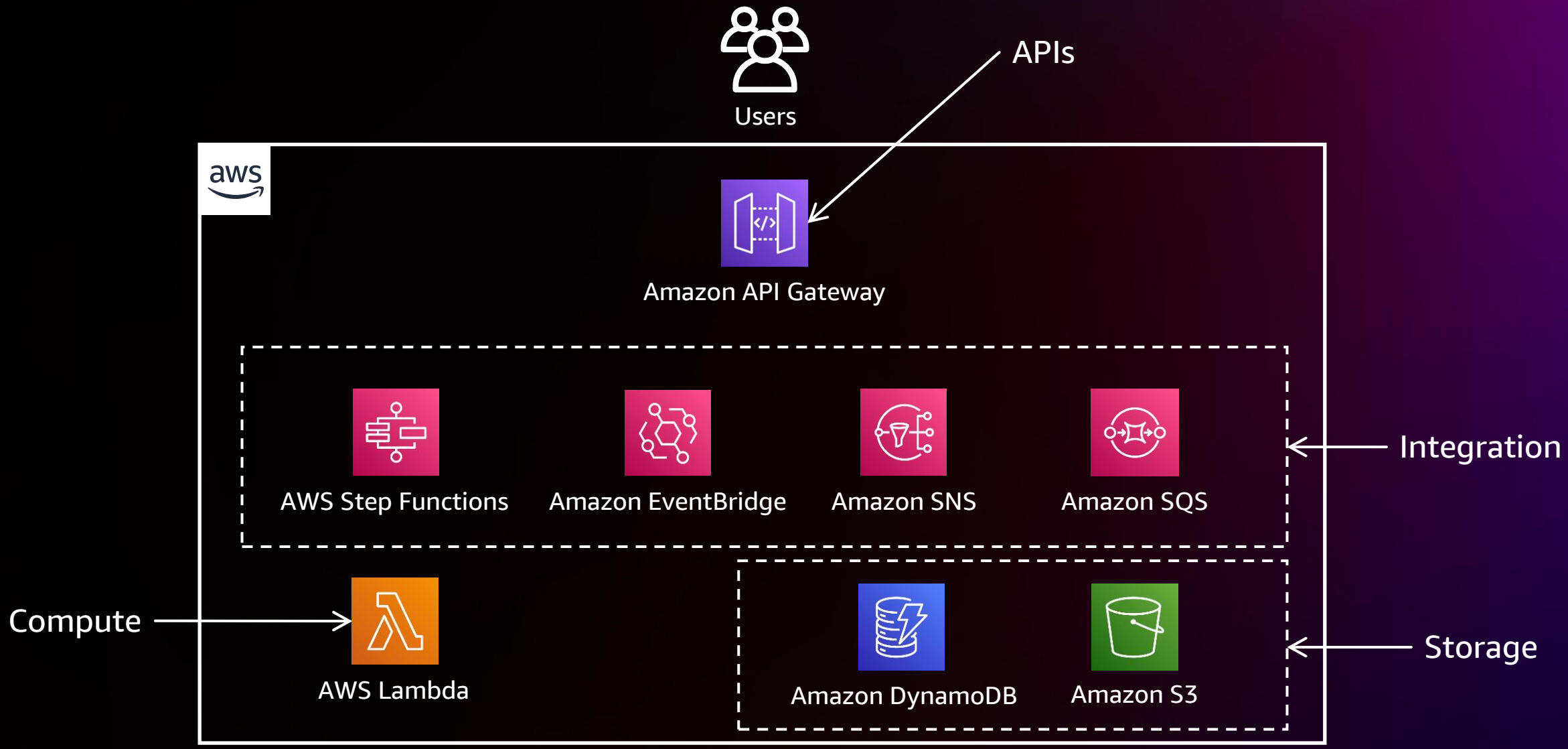

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The serverless toolbox



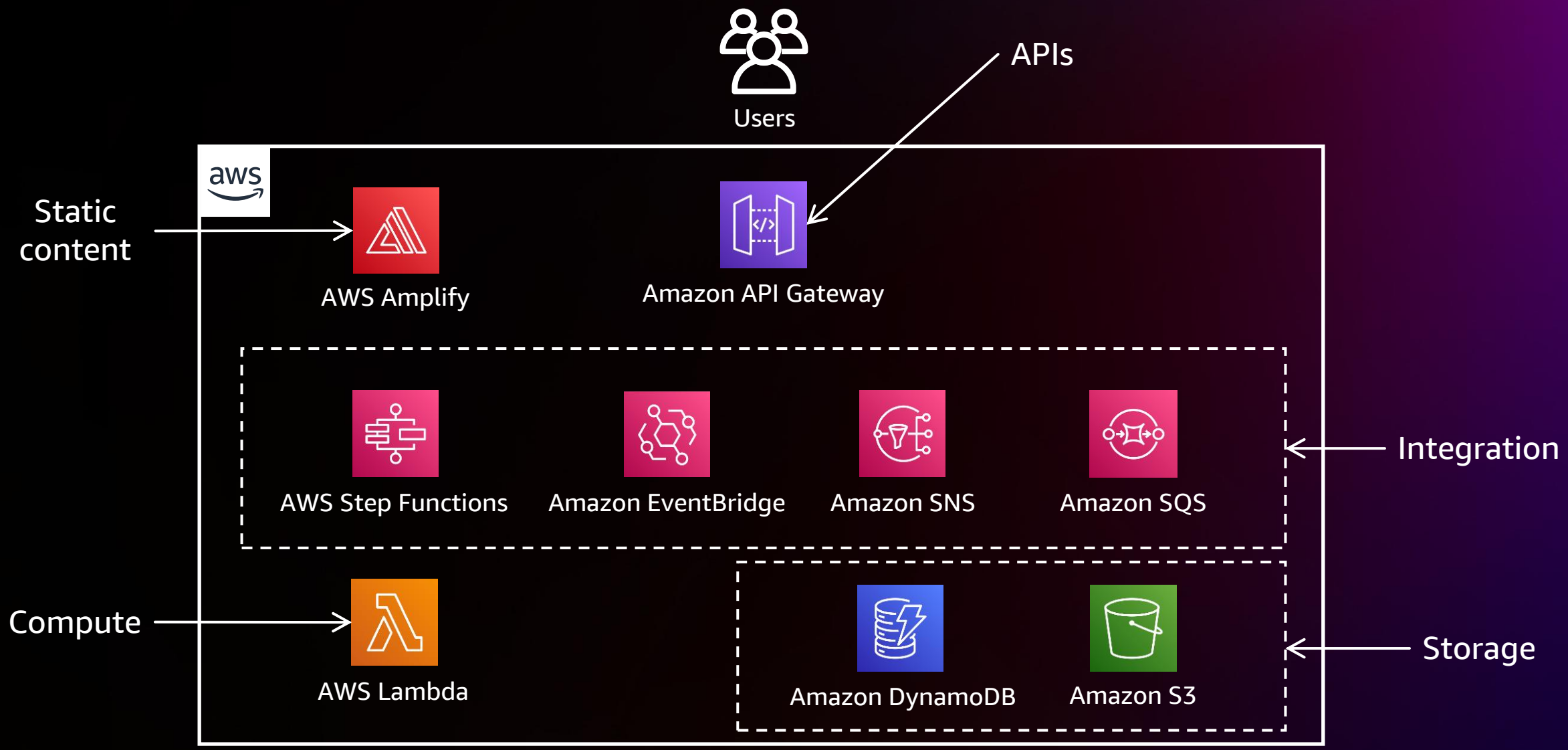

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The serverless toolbox



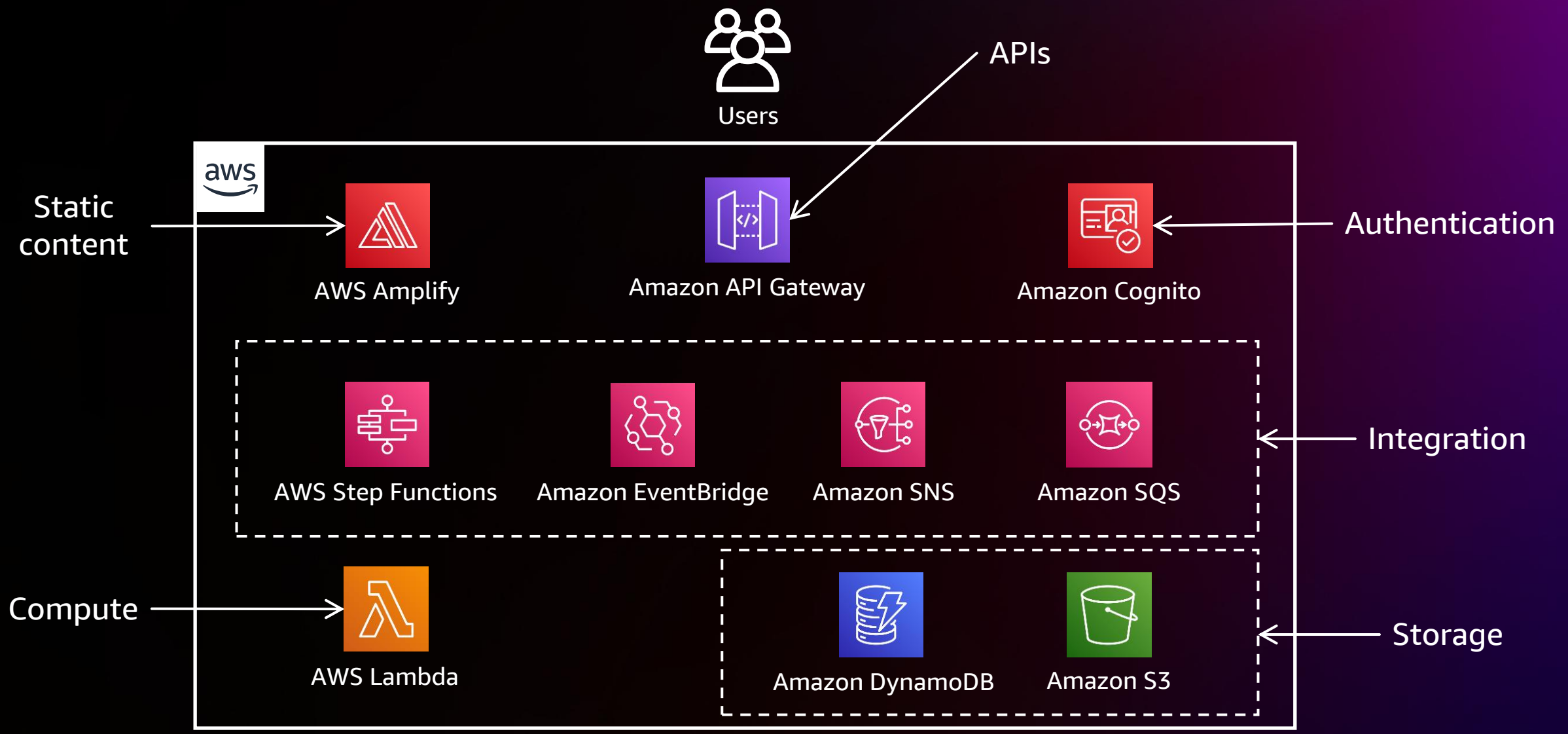

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# The serverless toolbox



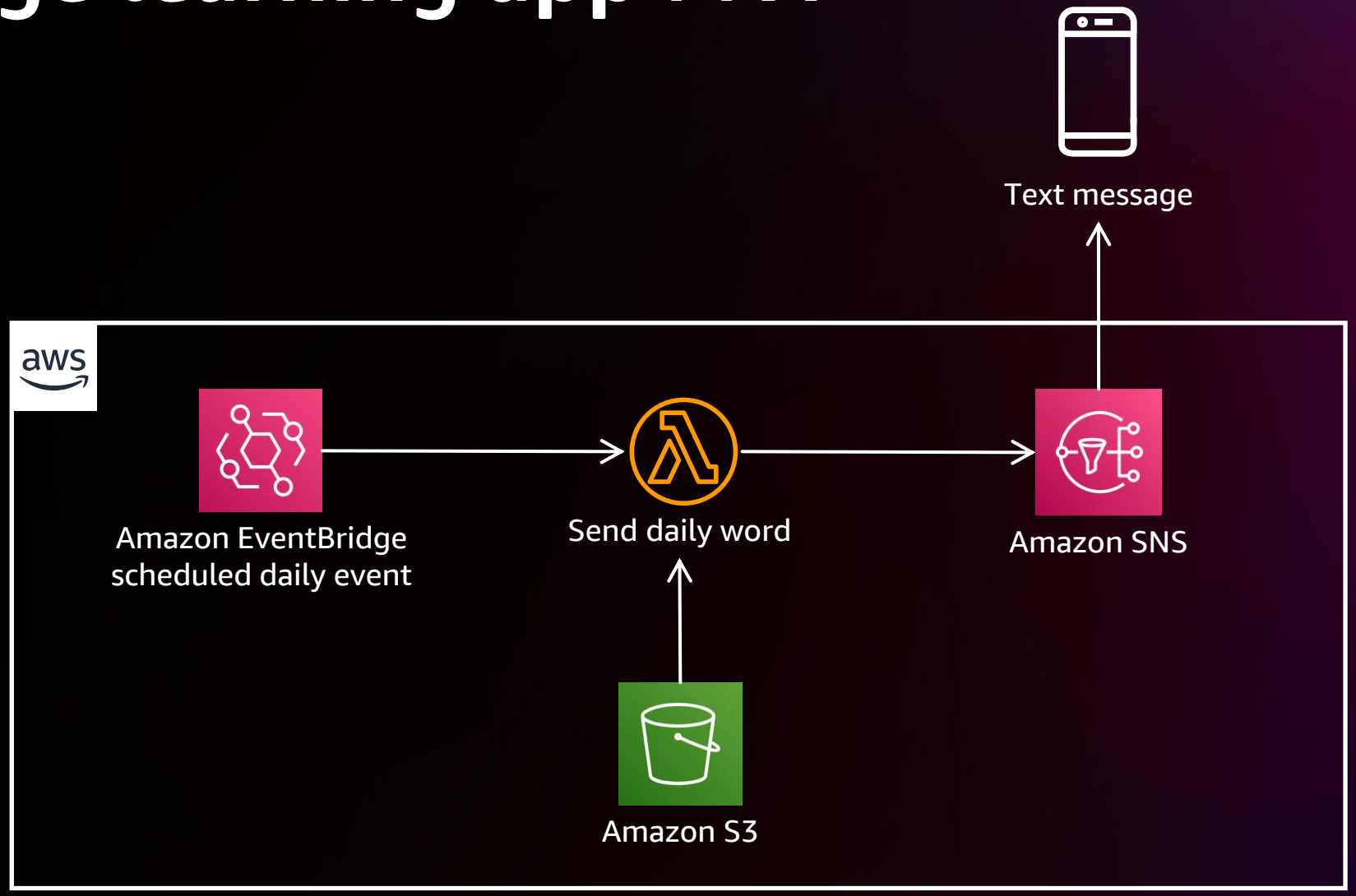

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Full-stack serverless application




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Language learning app MVP




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Example function code: Send text message

```
import json
import boto3

s3_client = boto3.client('s3')
sns_client = boto3.client('sns')

def lambda_handler(event, context):

  # Call S3 to retrieve today's vocabulary word

  message = “Today’s word is ” + word

  try:
    response = sns_client.publish(
      TargetArn='arn:aws:sns:us-east-1:1234567890:App',
      Message=json.dumps({'default': json.dumps(message)}),
      MessageStructure='json'
    )

  except Exception as e:
     print(f"Failed to send text message. Error: {e}")
```

aws

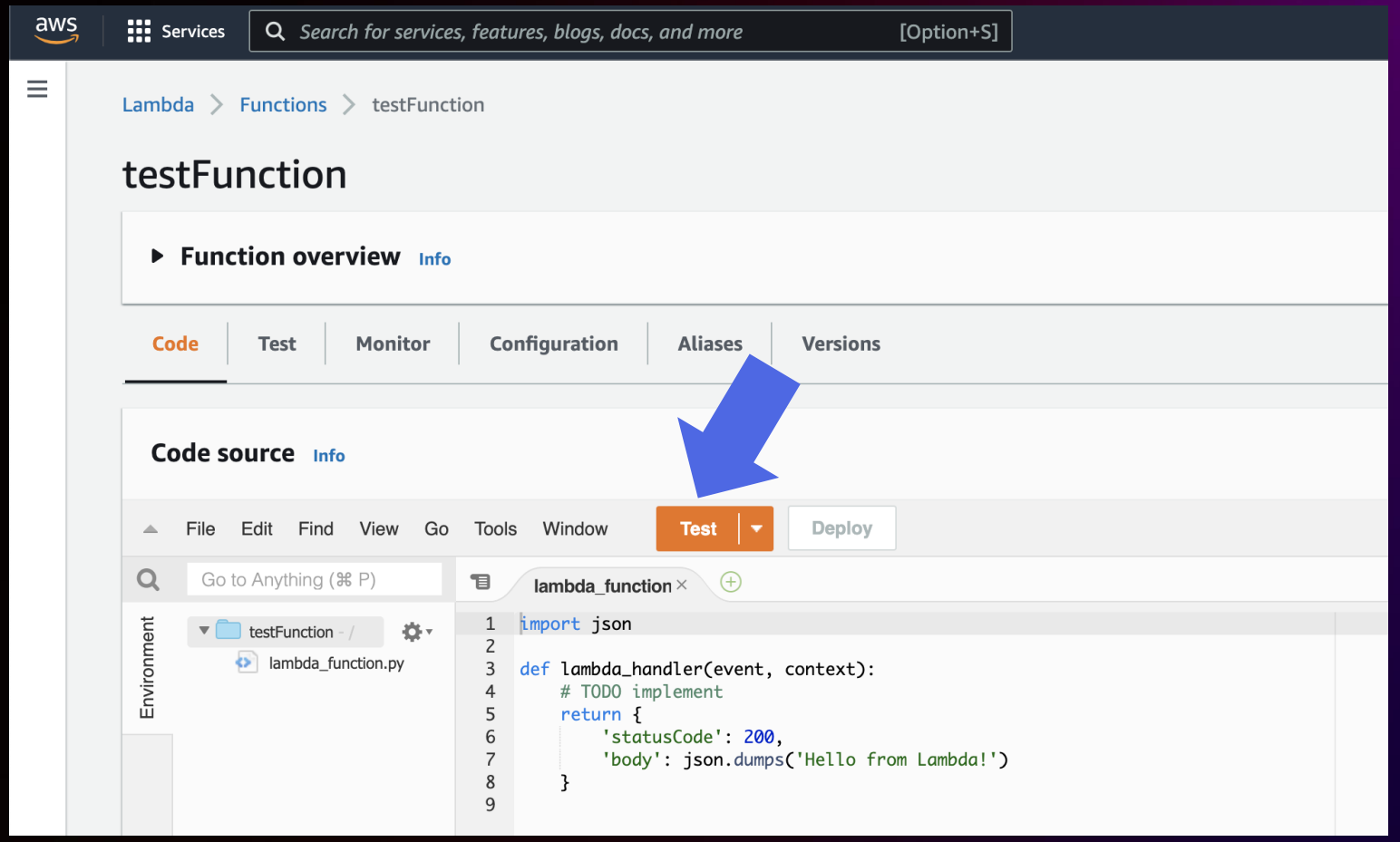
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Step 3: Development best practices

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Developer tools

- Experiment in the console to get started




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Developer tools: IaC framework

- Experiment in the console to get started
- Deploy using an infrastructure as code (IaC) framework






serverless



aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Example AWS SAM template

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: 'AWS::Serverless-2016-10-31'
Description: An app that sends a daily text message with a Chinese vocab word.

Resources:
  SendTextFunction:
    Type: 'AWS::Serverless::Function'
    Properties:
    Handler: app.lambda_handler
    Runtime: python3.8
    Policies:
     - SNSPublishMessagePolicy:
         TopicName: !GetAtt SendTextSNSTopic.TopicName
    Environment:
      Variables:
        SNS_TOPIC_ARN: !Ref SendTextSNSTopic
    Events:
      DailyEventBridgeEvent:
        Type: Schedule
        Properties:
          Schedule: cron(0 20 * * ? *)

# S3 Bucket and SNS Topic defined below
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Developer tools: CI/CD

- Experiment in the console to get started
- Deploy using an infrastructure as code (IaC) framework
- Set up an automated deployment pipeline

circleci



AWS CodePipeline

GitHub

*This is not a complete list. To view all AWS Partners for this category, visit AWS Partner Solutions Finder. This list of partners is current as of October 17, 2022.*

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Automated deployment pipeline




Application is defined using a serverless development framework



AWS CodeCommit

Code is uploaded to a repository



AWS CodePipeline

New code launches a new build process



New application code is deployed

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Developer tools: Testing

## Two approaches to testing

- For testing complex Lambda function code logic → use unit tests

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Example function code unit test

```
import unittest
from unittest import mock
from my_function.app import lambda_handler

def mocked_read_s3_file(bucket_name, key_name):
  return # Mocked file contents from S3
def mocked_publish_message():
  return # Mocked SNS topic response

class MyFunctionTest(unittest.TestCase):

  @mock.patch('my_function.app.read_s3_file', side_effect=mocked_read_s3_file)
  @mock.patch('my_function.app.publish_sns_message', side_effect=mocked_publish_sns_message)

  def test_build(self, read_s3_file_mock, publish_sns_message_mock):
    response = lambda_handler(self.test_event(), "")
    self.assertEqual(read_s3_file_mock.call_count, 1)
    self.assertEqual(publish_sns_message_mock.call_count, 1)

def test_event(self):
  return # Mocked EventBridge scheduled event payload
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Developer tools: Testing

## Two approaches to testing

- For testing complex Lambda function code logic → use unit tests
- For testing service integrations → test in the cloud

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Test in the cloud faster

- AWS SAM Accelerate lets you quickly iterate against the cloud
- Sync code changes using the sam sync CLI command
- Useful for development environments

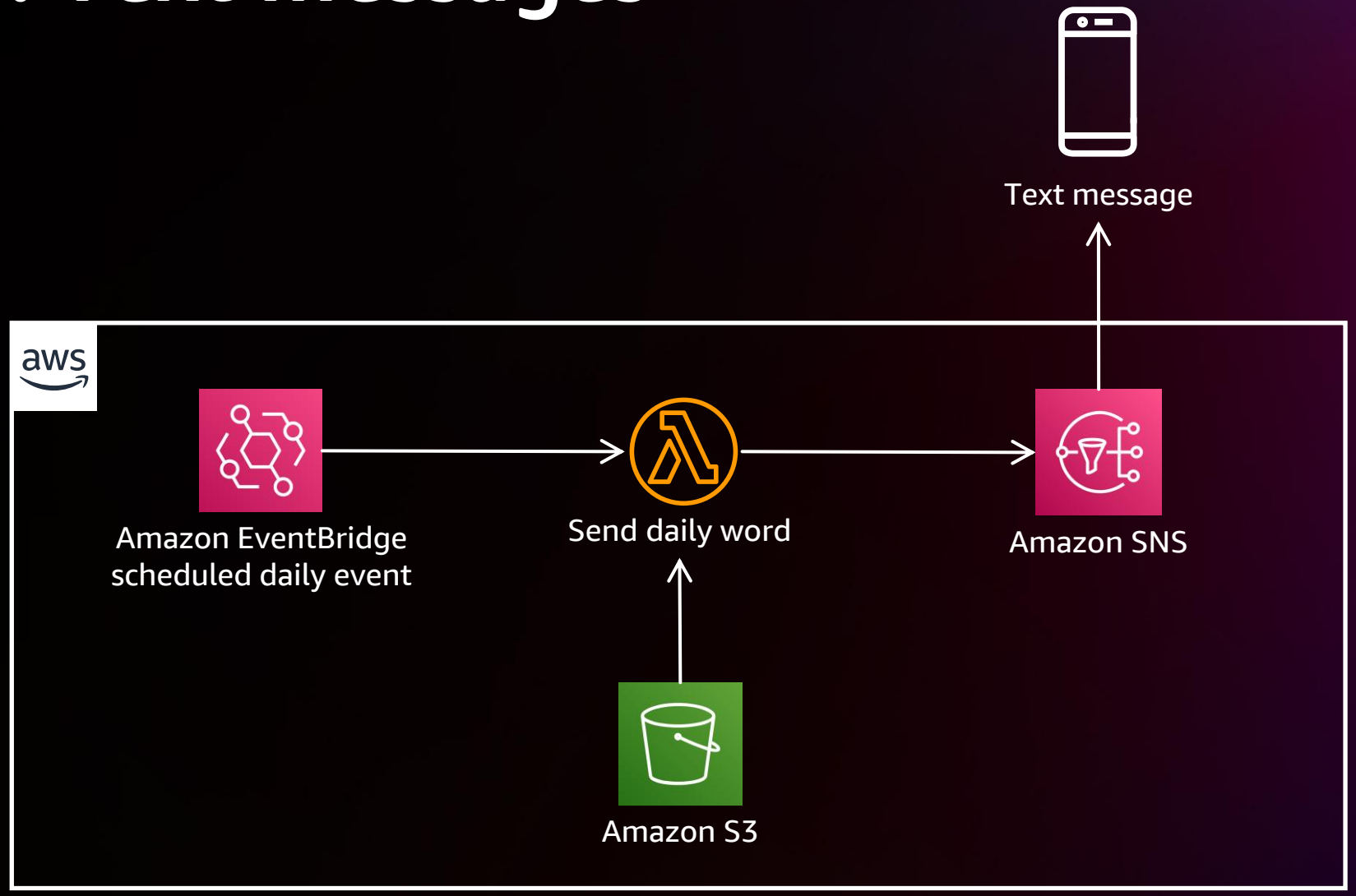


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Step 4: Building the app

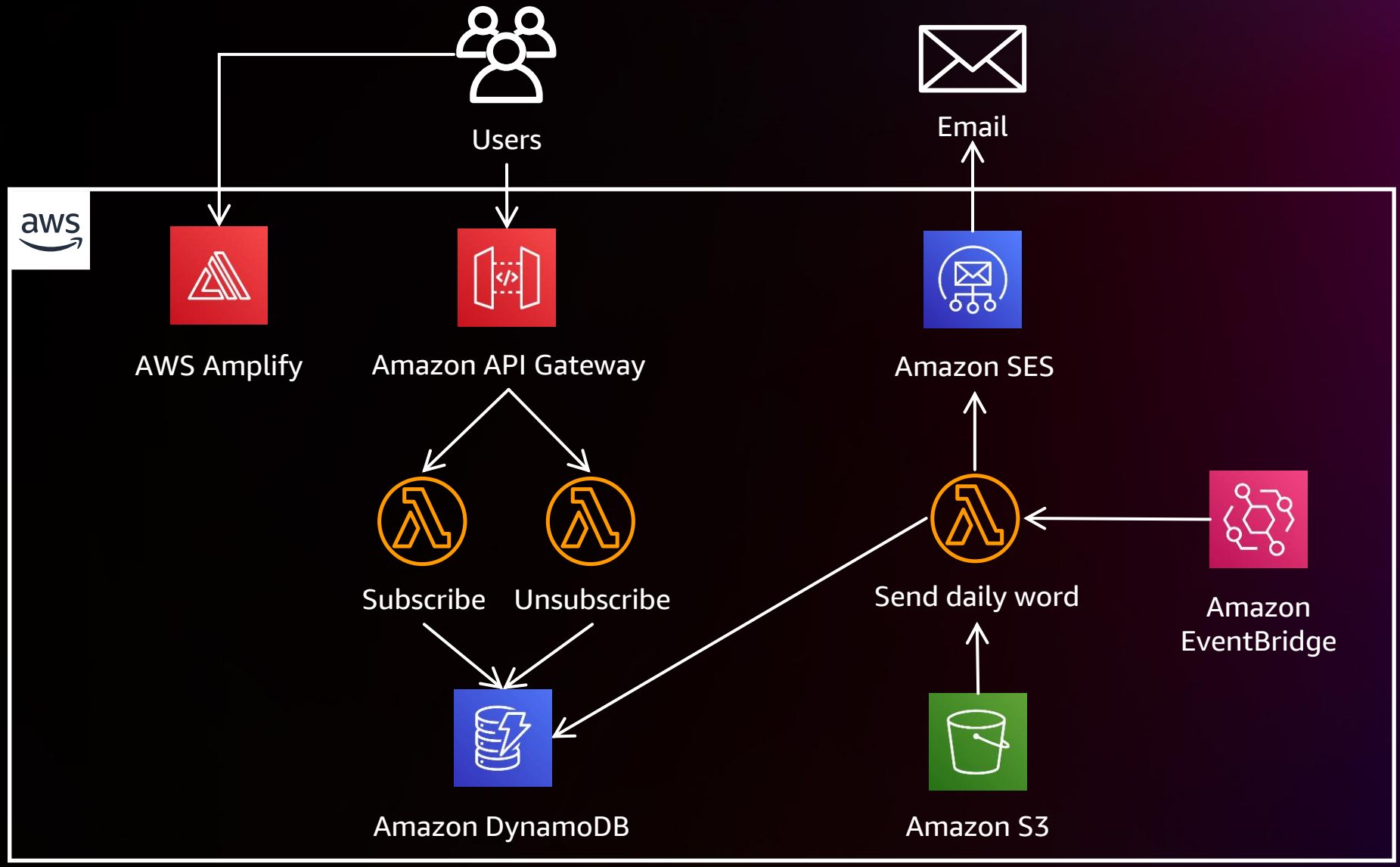
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phase 1: Text messages

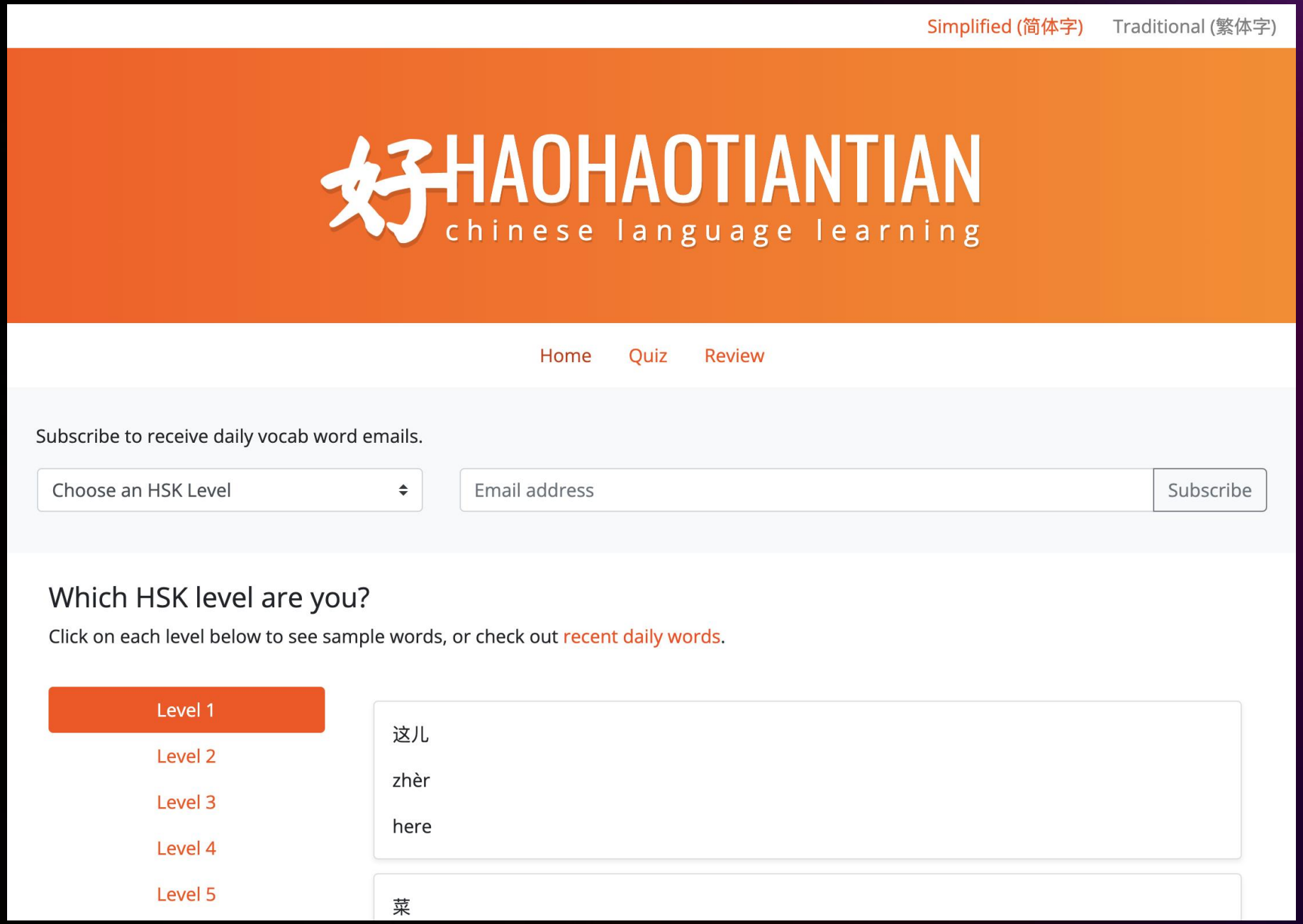



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Phase 2: Subscribers, APIs, and a static website



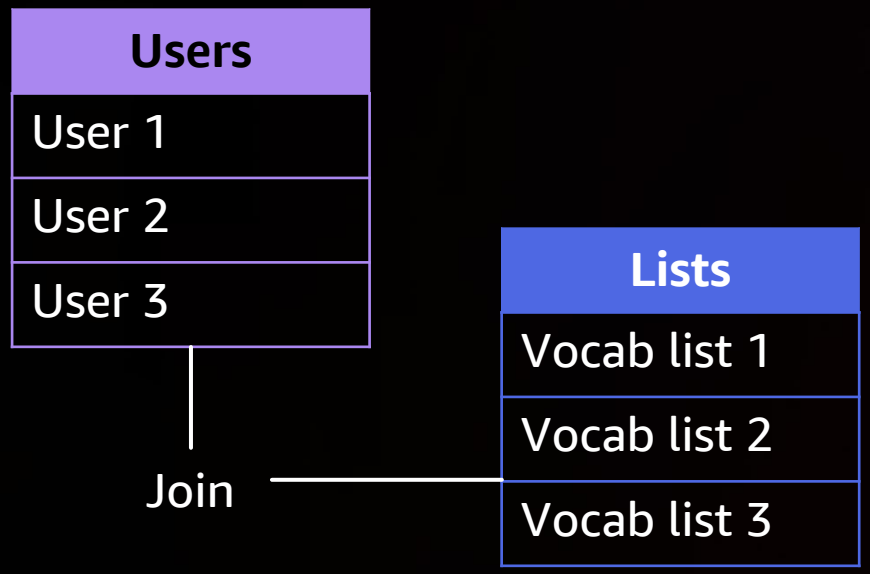

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Plan your DynamoDB access patterns

Typical SQL data structure

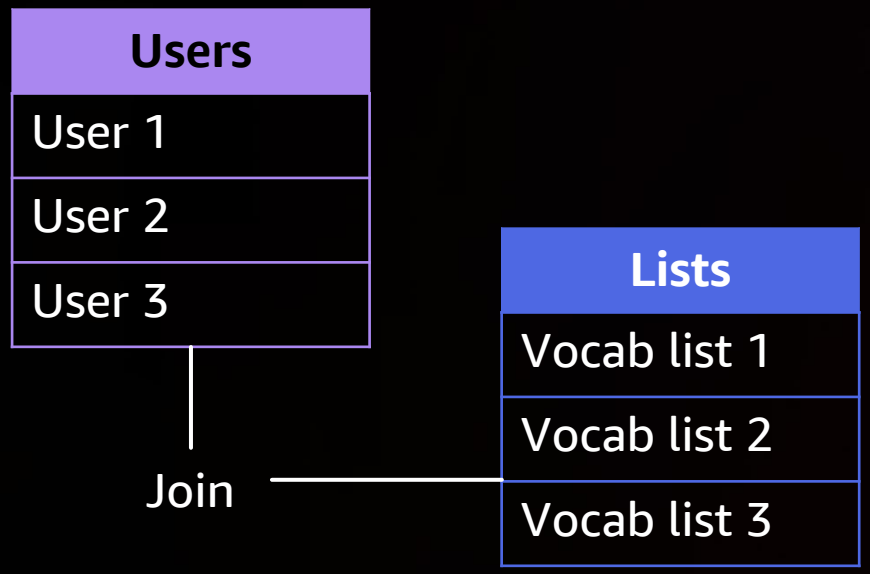



SQL uses **joins** and **aggregates** data well

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Plan your DynamoDB access patterns

## Typical SQL data structure




## NoSQL data structure

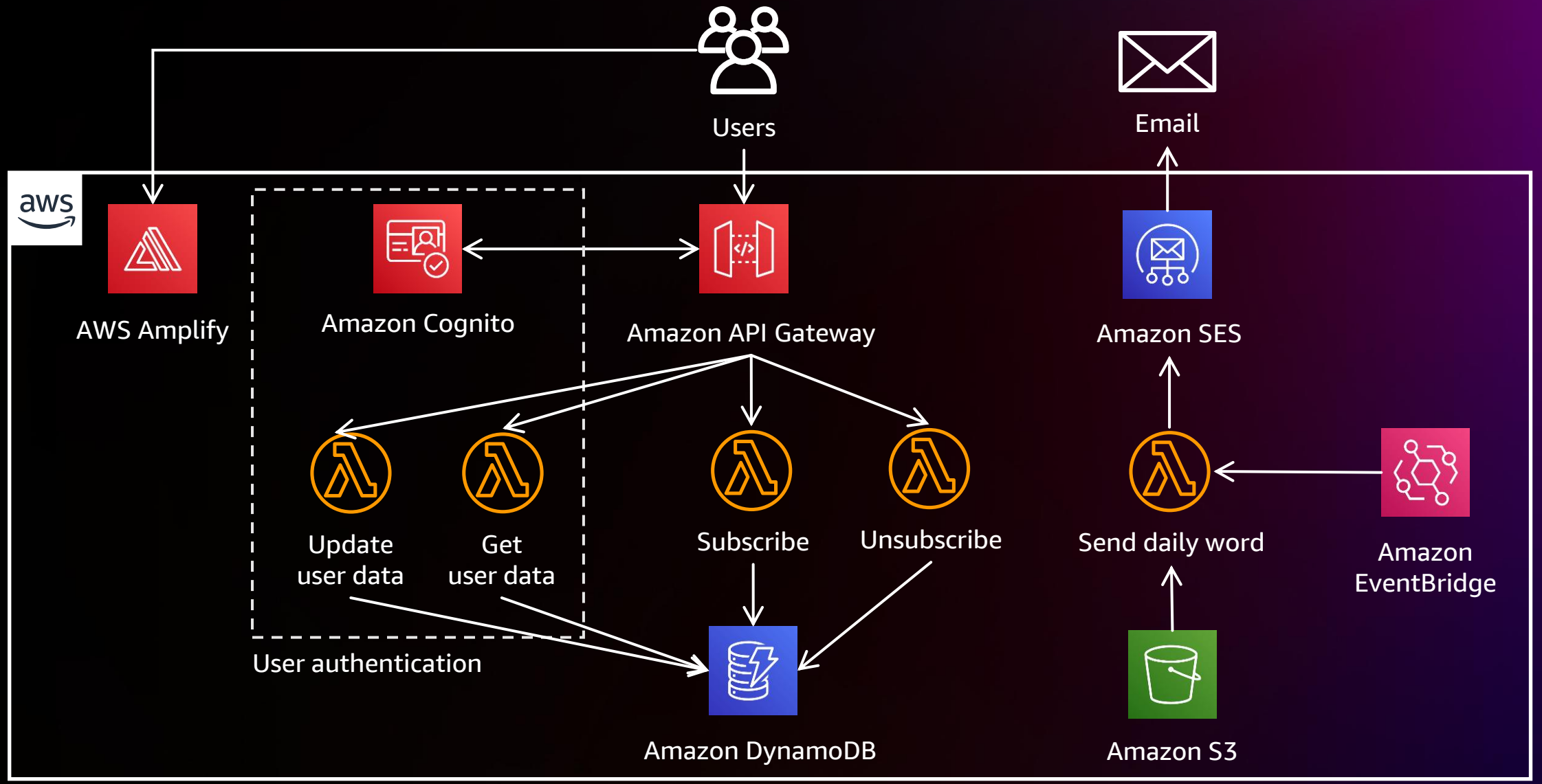
| Users and lists | |
|---|---|
| User 1 | Vocab list 1 |
| | Vocab list 2 |
| User 2 | Vocab list 1 |
| User 3 | Vocab list 2 |
| | Vocab list 3 |

SQL uses **joins** and **aggregates** data well

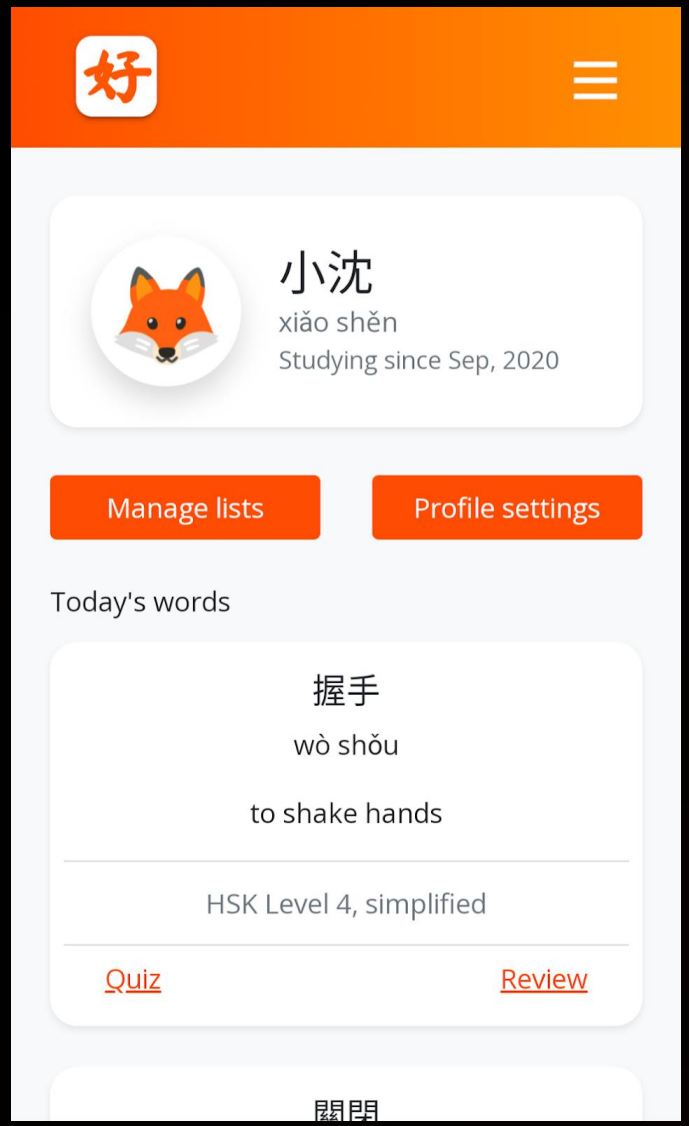
NoSQL uses **item collections** and is designed for **low latency** at scale

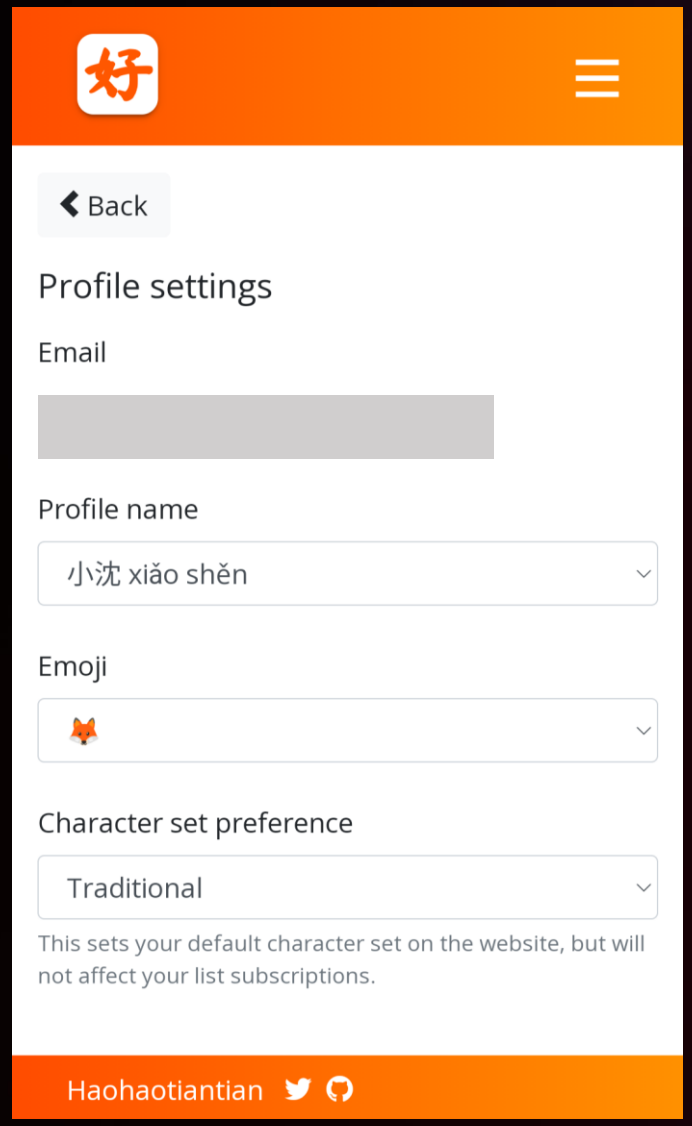
aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

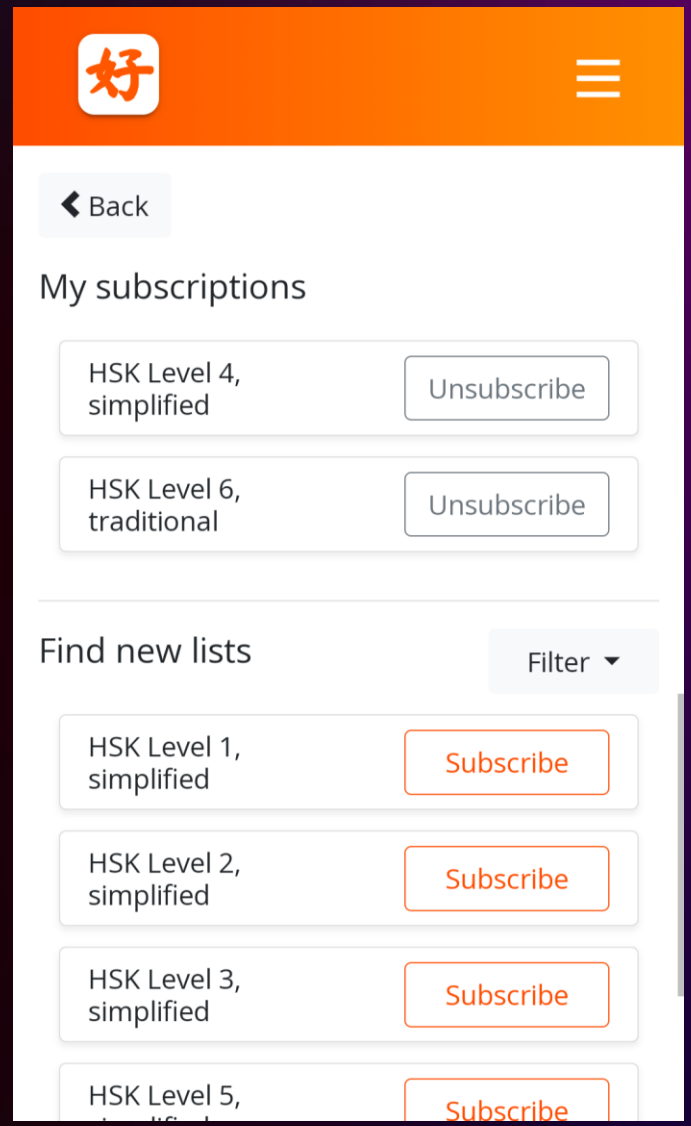
# Phase 3: User login and profiles




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



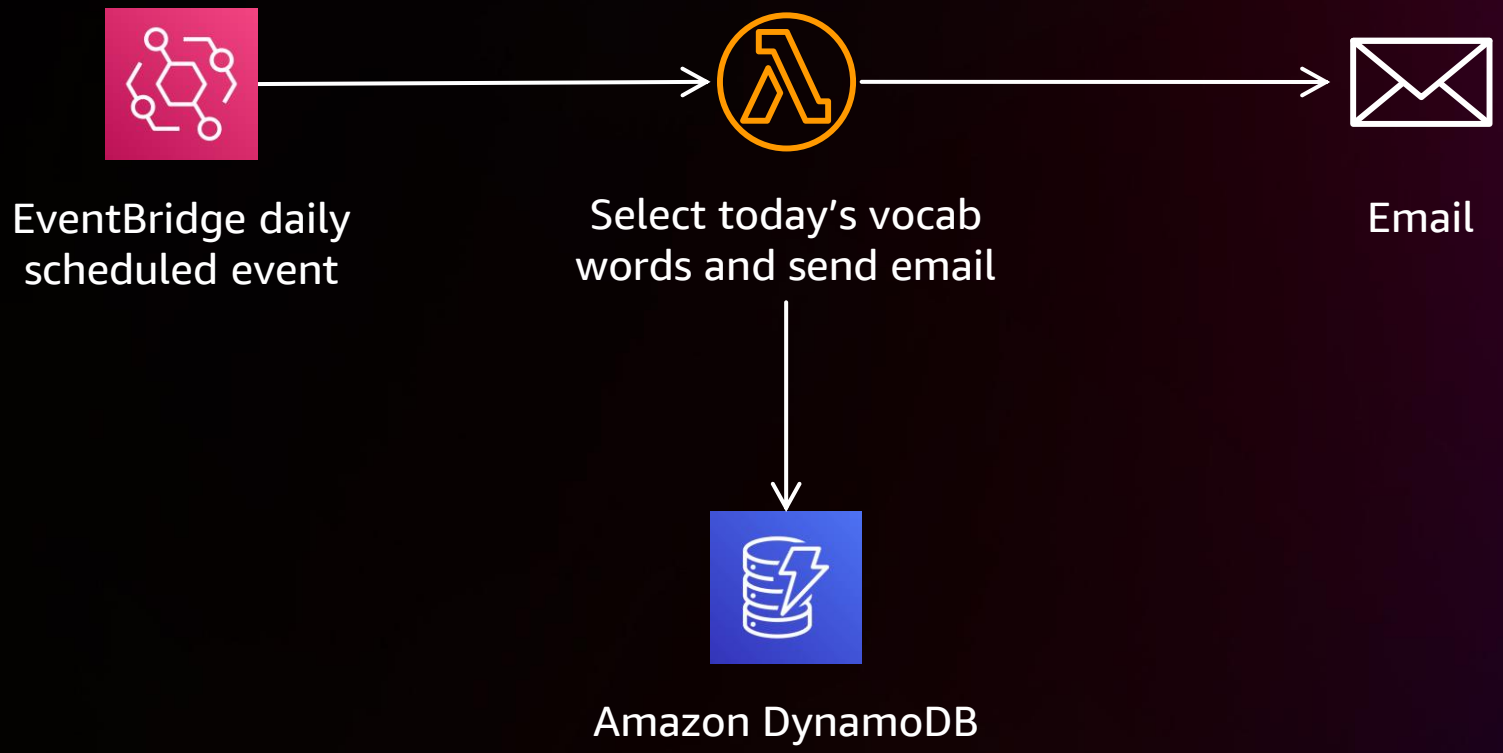




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Step 5: Extending the app

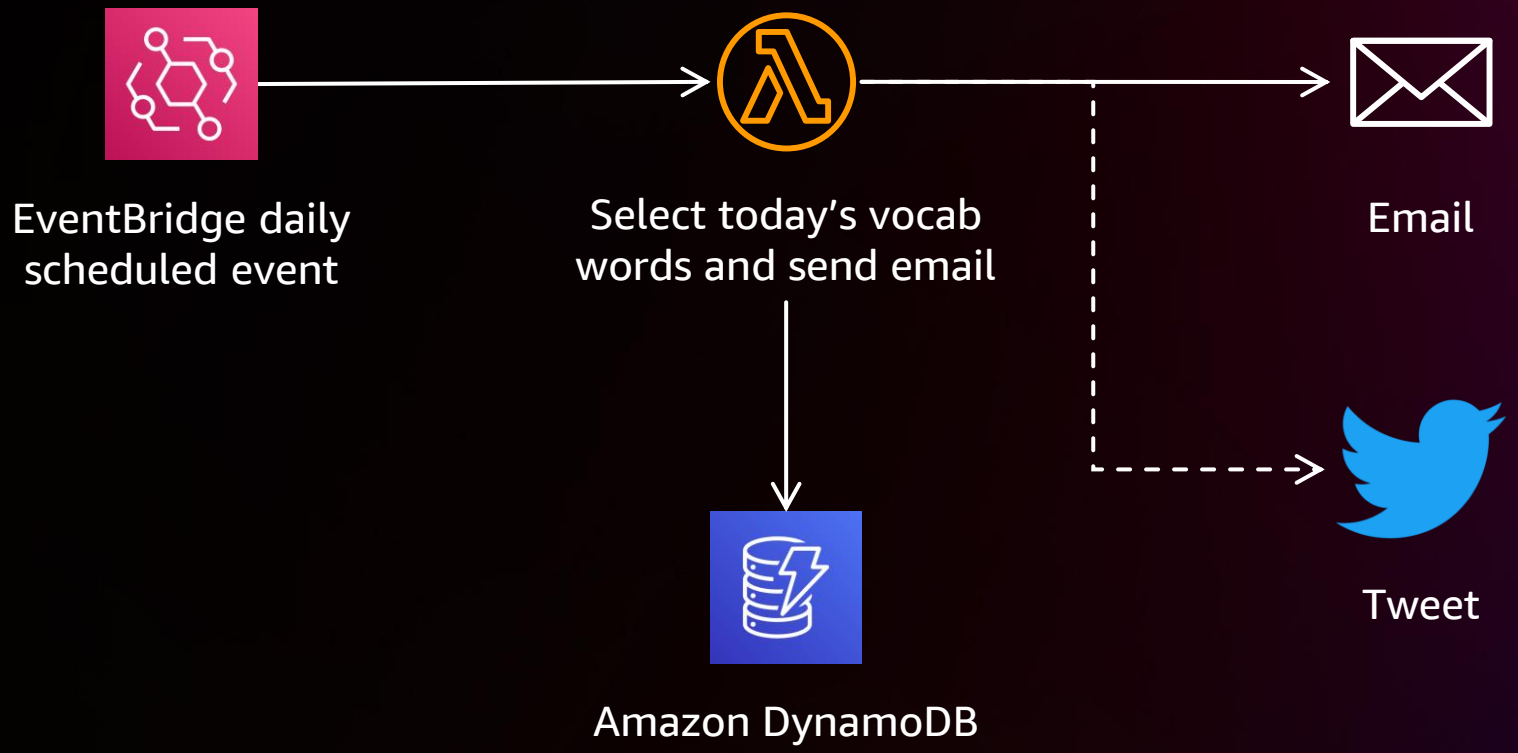
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Daily vocab tweet bot



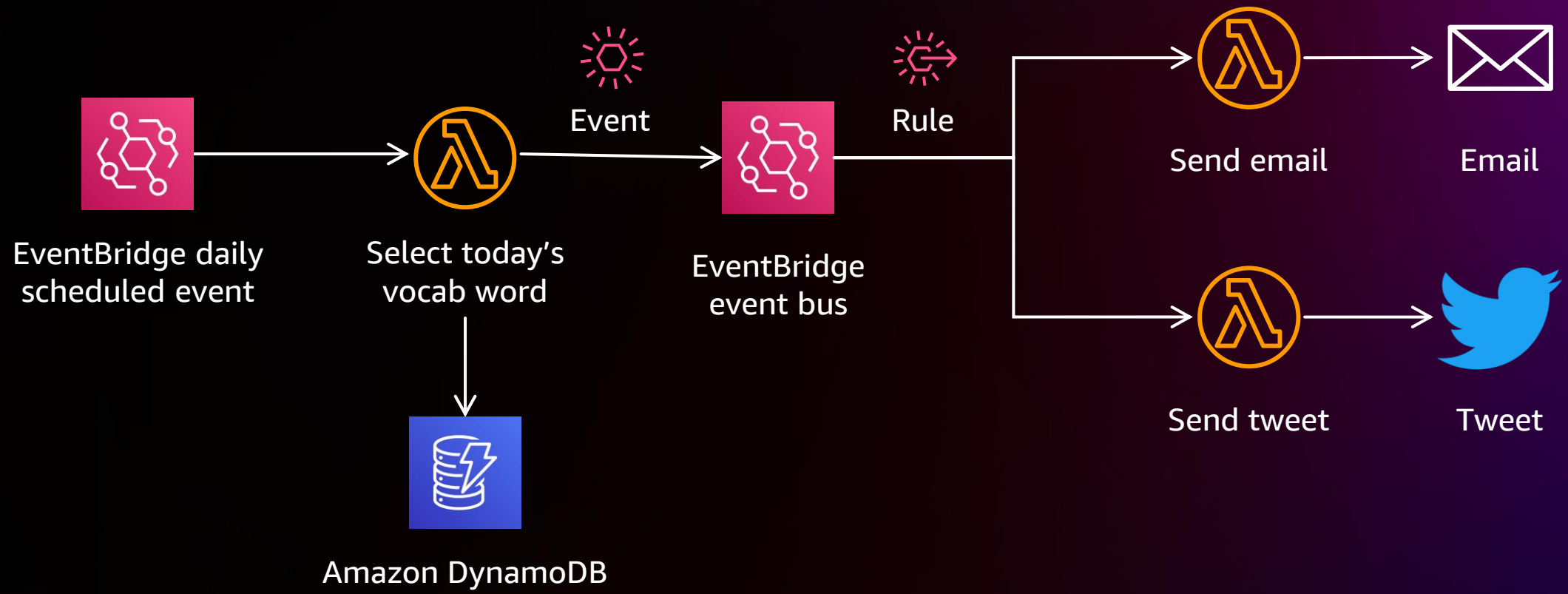

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Daily vocab tweet bot



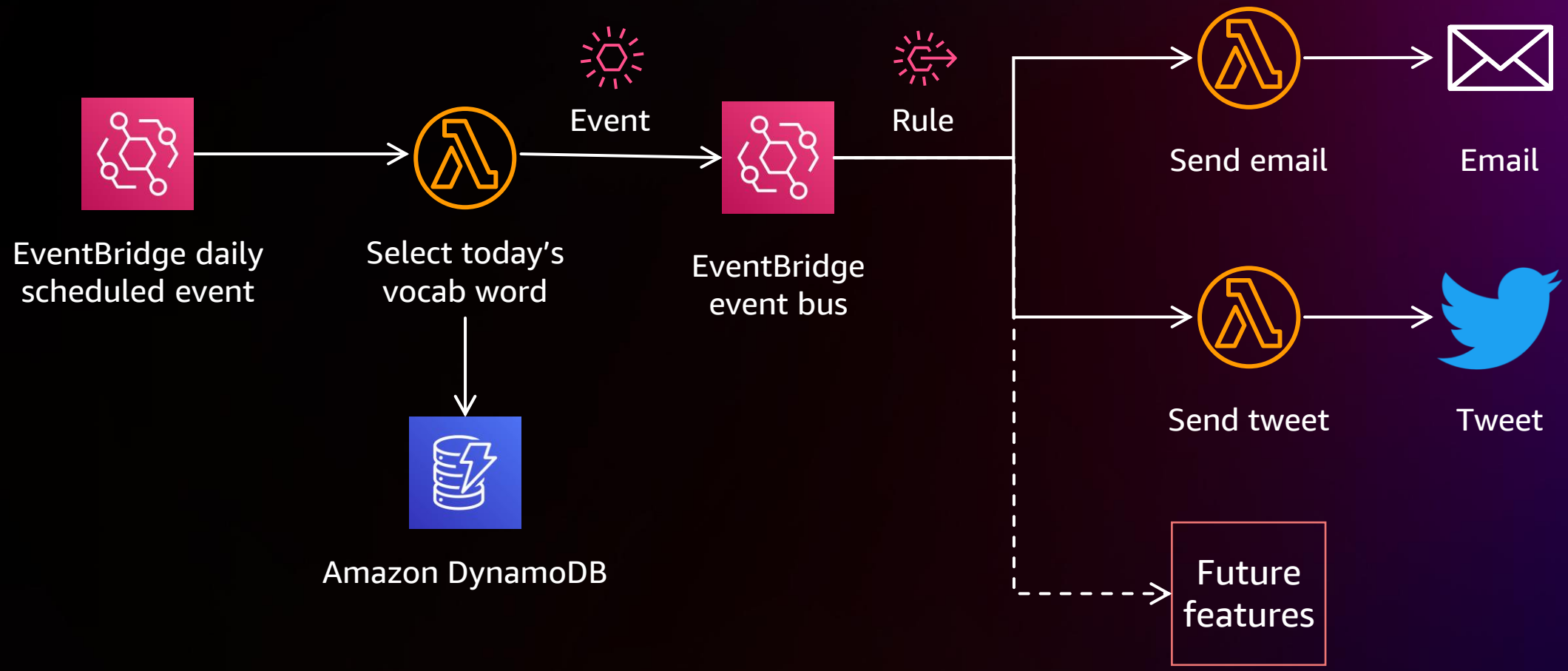

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Daily vocab tweet bot




aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Daily vocab tweet bot




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Daily vocab tweet bot

Event schema

```
{
    “source”: “vocab-app”,
    “detail-type”: “todays-words-set”,
    “detail”: {
        “idempotency-key”: “0123456”
    }
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Daily vocab tweet bot

Event schema

```
{
    “source”: “vocab-app”,
    “detail-type”: “todays-words-set”,
    “detail”: {
        “idempotency-key”: “0123456”
    }
}
```

“The vocab words for today have been selected”

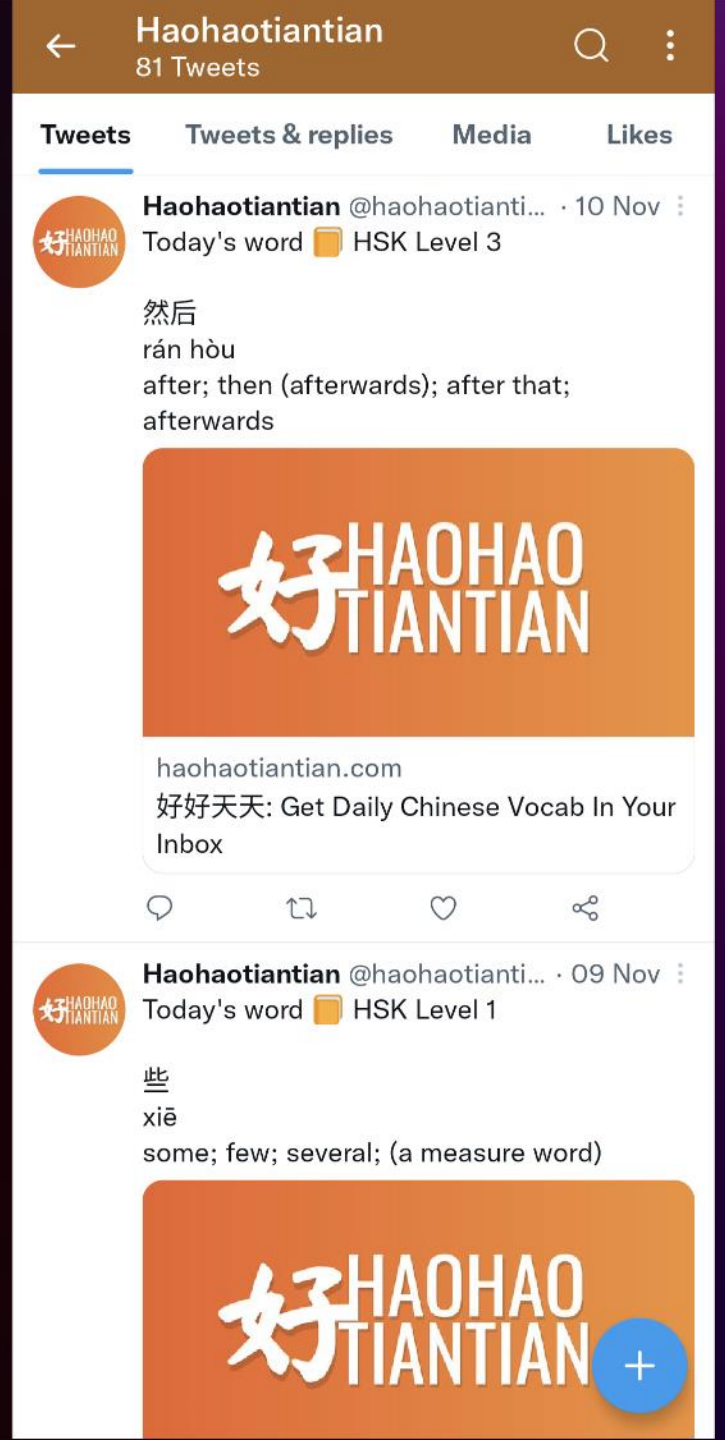
Unique identifier so that services processing the event don’t process it twice

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

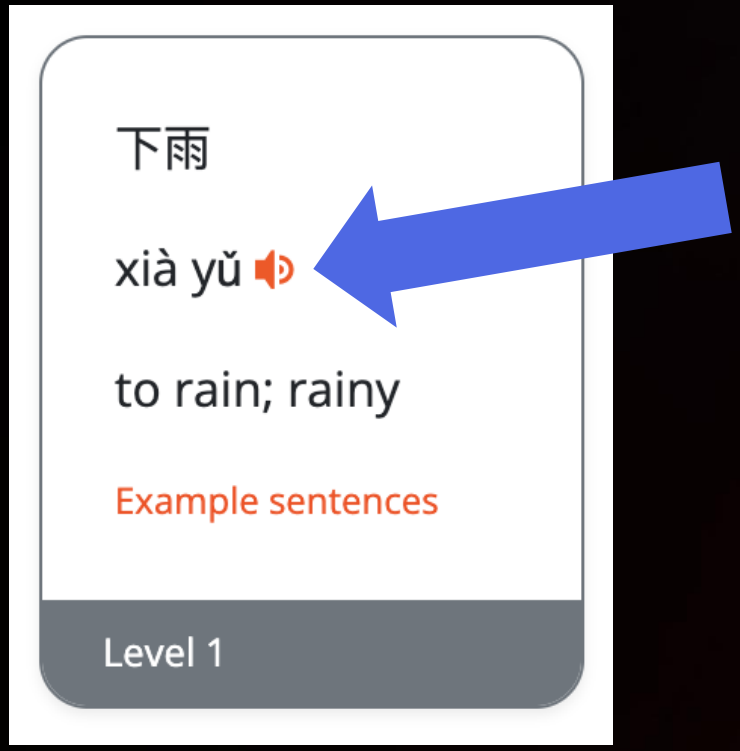
# New feature: Daily vocab tweet bot

@haohaotiantian_




aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Vocab word audio

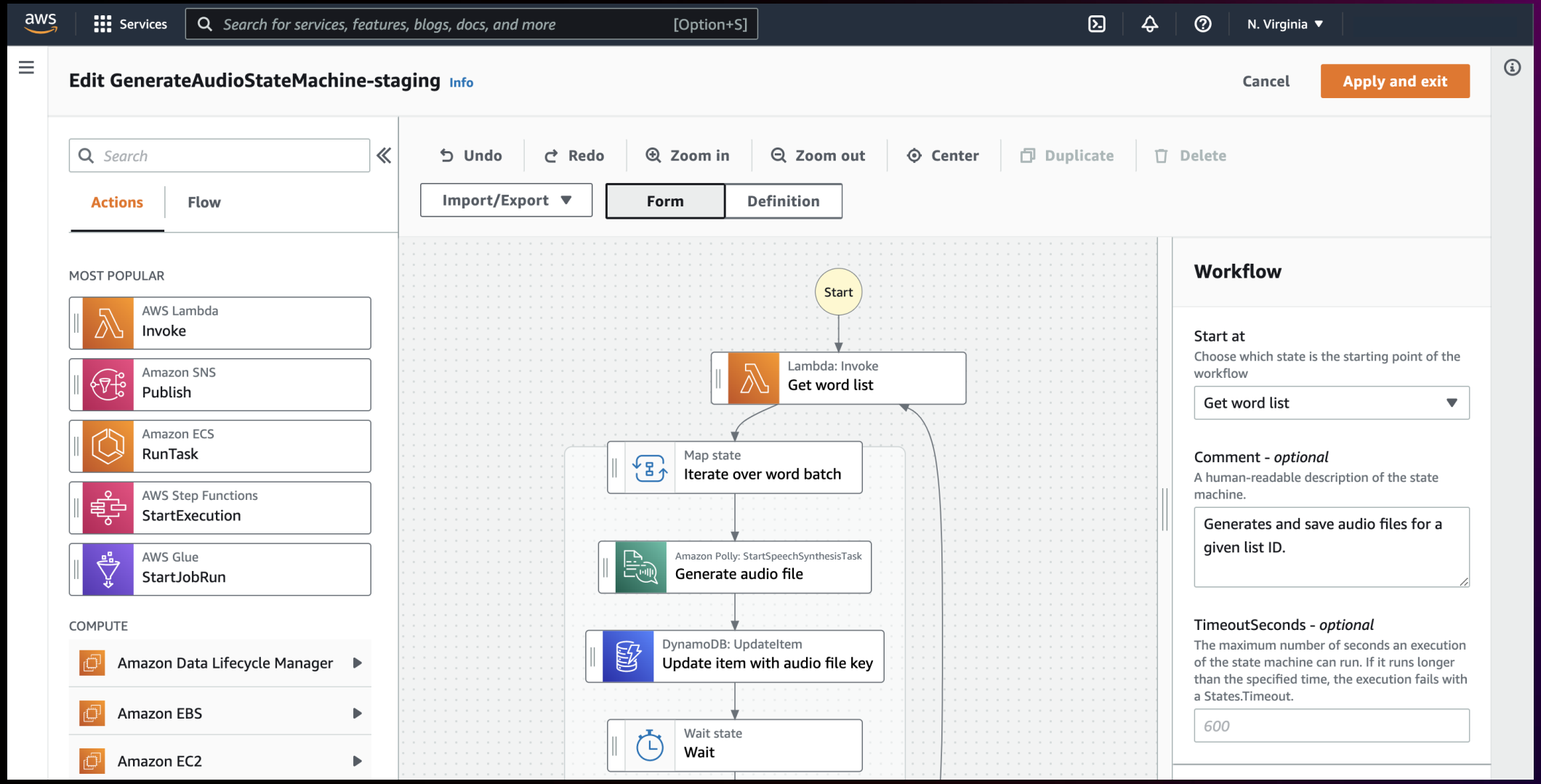





Amazon Polly

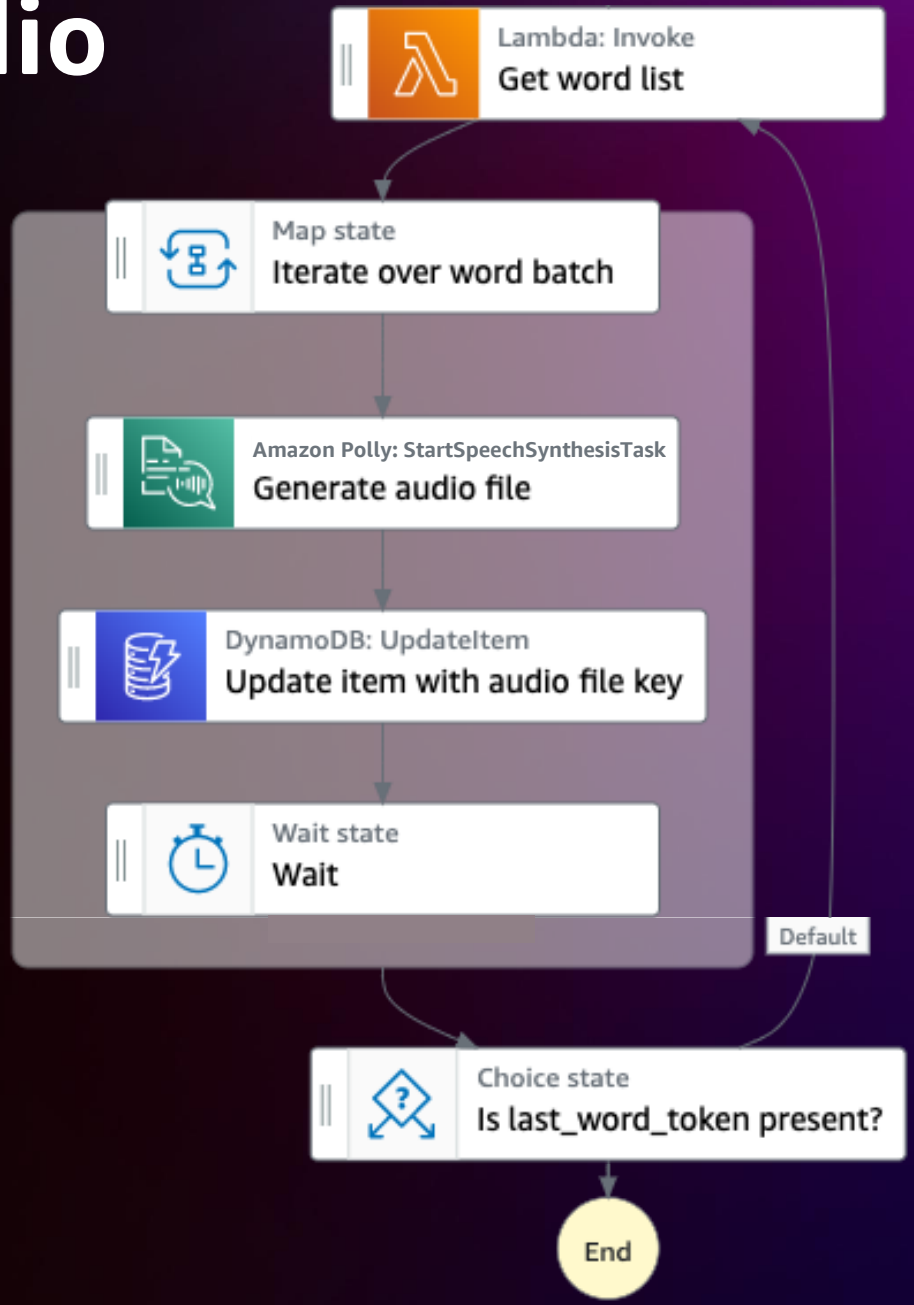
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Step Functions Workflow Studio




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Vocab word audio



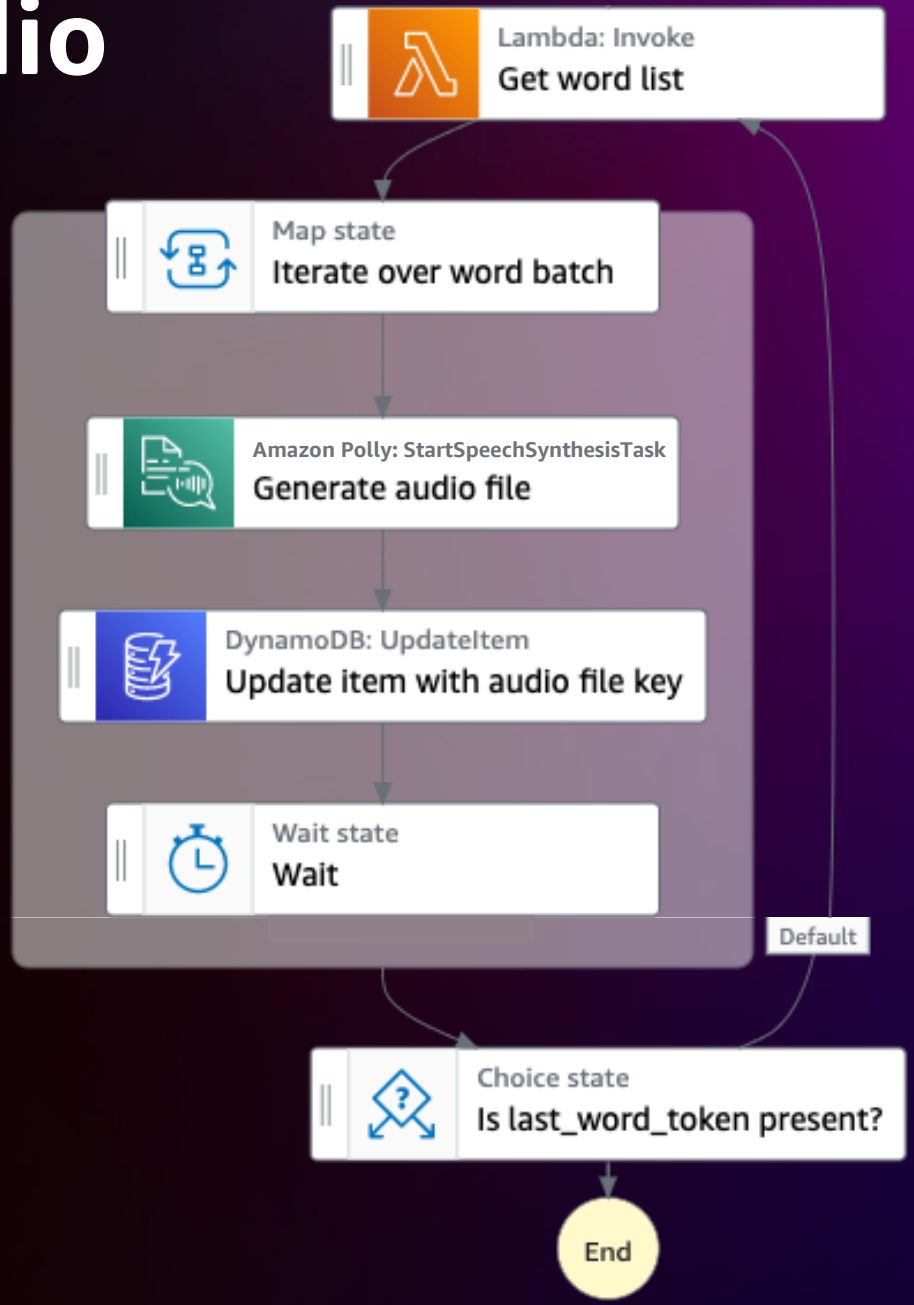

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# New feature: Vocab word audio

Loop through batches of vocab words

Step Functions AWS SDK integration calls services APIs

Wait state to rate limit calls to Amazon Polly API




© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Monthly cost

- 100+ active subscribers
- 4.5 thousand API Gateway requests
- 4.8 thousand Lambda invokes
- 10 thousand Amazon CloudFront requests
- 32 thousand Amazon S3 requests



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Application monthly cost

Free Tier

| Service | Cost |
|---|---|
| Amazon Route 53 hosted zone | $0.50 |
| CloudFront | $0.11 |
| Amazon S3 | $0.08 |
| API Gateway | $0.01 |
| EventBridge | $0.00 |
| Amazon Cognito | $0.00 |
| DynamoDB | $0.00 |
| Lambda | $0.00 |
| Step Functions | $0.00 |
| Amazon SES | $0.00 |
| Tax | $0.12 |
| **Total monthly cost** | **$0.82** |

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Tips for building your first serverless application

Developer tooling

- Deploy using an IaC framework
- Set up an automated deployment pipeline
- Test in the cloud

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Tips for building your first serverless application

## Developer tooling

- Deploy using an IaC framework
- Set up an automated deployment pipeline
- Test in the cloud

## Application design

- Plan your database access patterns
- Use workflows to orchestrate services
- Emit events to extend your app

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

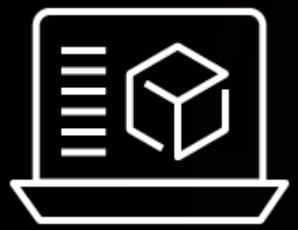
Serverless and event-driven architectures let you
**turn your ideas into applications faster**

@em__shea
https://haohaotiantian.com
https://github.com/em-shea/vocab

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Continue your AWS serverless learning

## Learn at your own pace



Expand your serverless skills with our Learning Plan on **AWS Skill Builder**

## Increase your knowledge



Use our **Ramp-Up Guides** to build your serverless knowledge

## Earn AWS serverless badges




Demonstrate your knowledge by achieving **digital badges**

https://s12d.com/serverless-learning

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Thank you!

Emily Shea
@em__shea



Please complete the session survey in the **mobile app**

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.