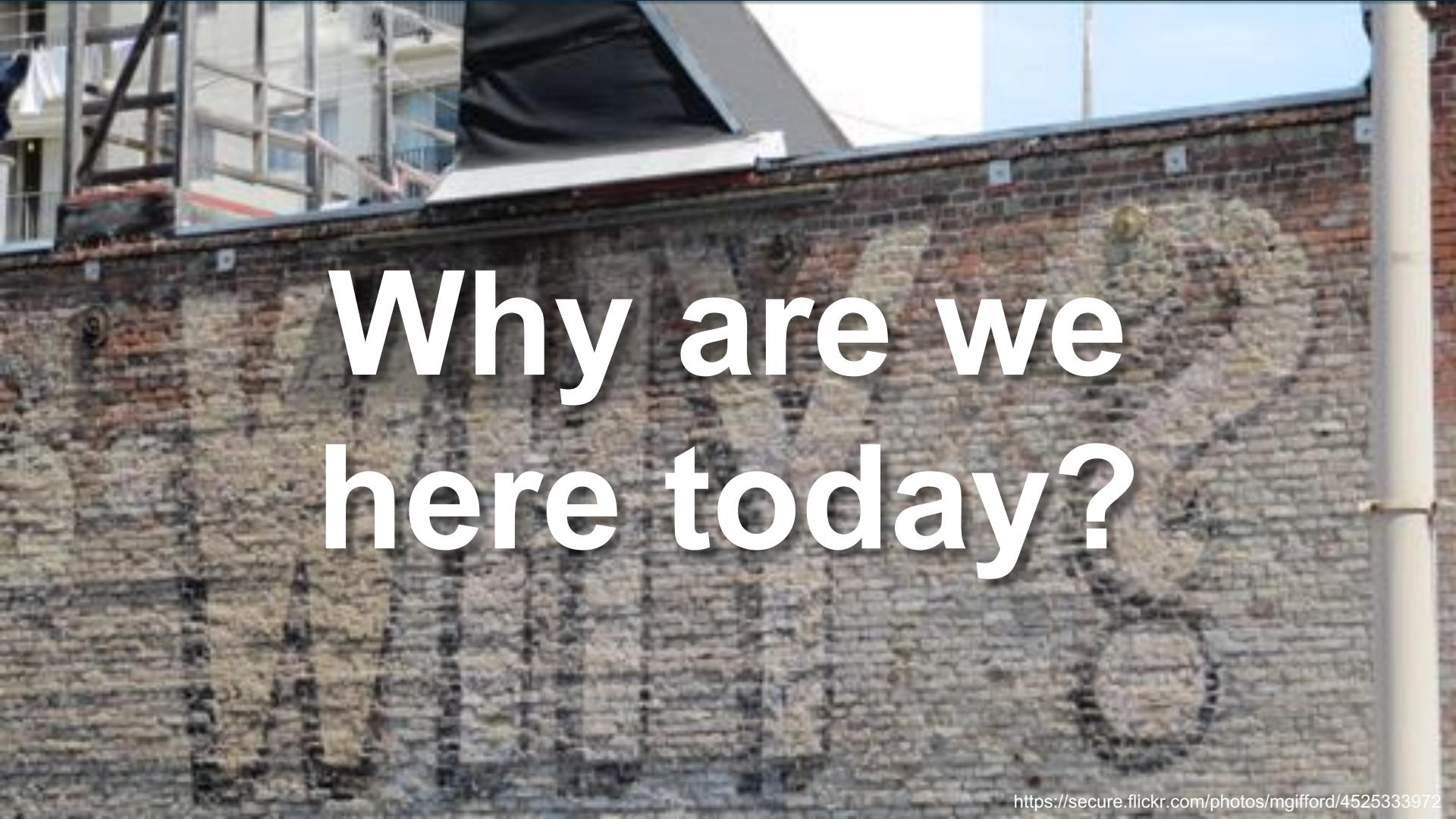
/serverless/ DAYS BOSTON 2019

How AWS builds Serverless services using Serverless

Chris Munns
Principal Developer Advocate
AWS Serverless

aws



https://secure.flickr.com/photos/mgifford/4525333972

Serverless is changing the software industry

Photo by Markus Spiske on Unsplash


Amazon is no different in this!
Photo by Chris Munns

So how does AWS build Serverless Services Using Serverless?

So how does AWS build Serverless Services Using Serverless?

With a compiler!

So how does AWS build Serverless Services Using Serverless?

With a compiler!

ba-dum-tshh

Photo by Josh Sorenson on Unsplash

Less time managing servers ==

more laughs!

For today

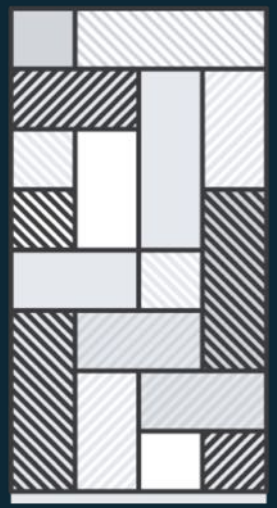
A look at how Amazon does development

How serverless has changed things

What you can learn from us

Development transformation at Amazon:

1994-2001



monolithic architecture +
hierarchical organization

2002+

decoupled services +
2 pizza teams

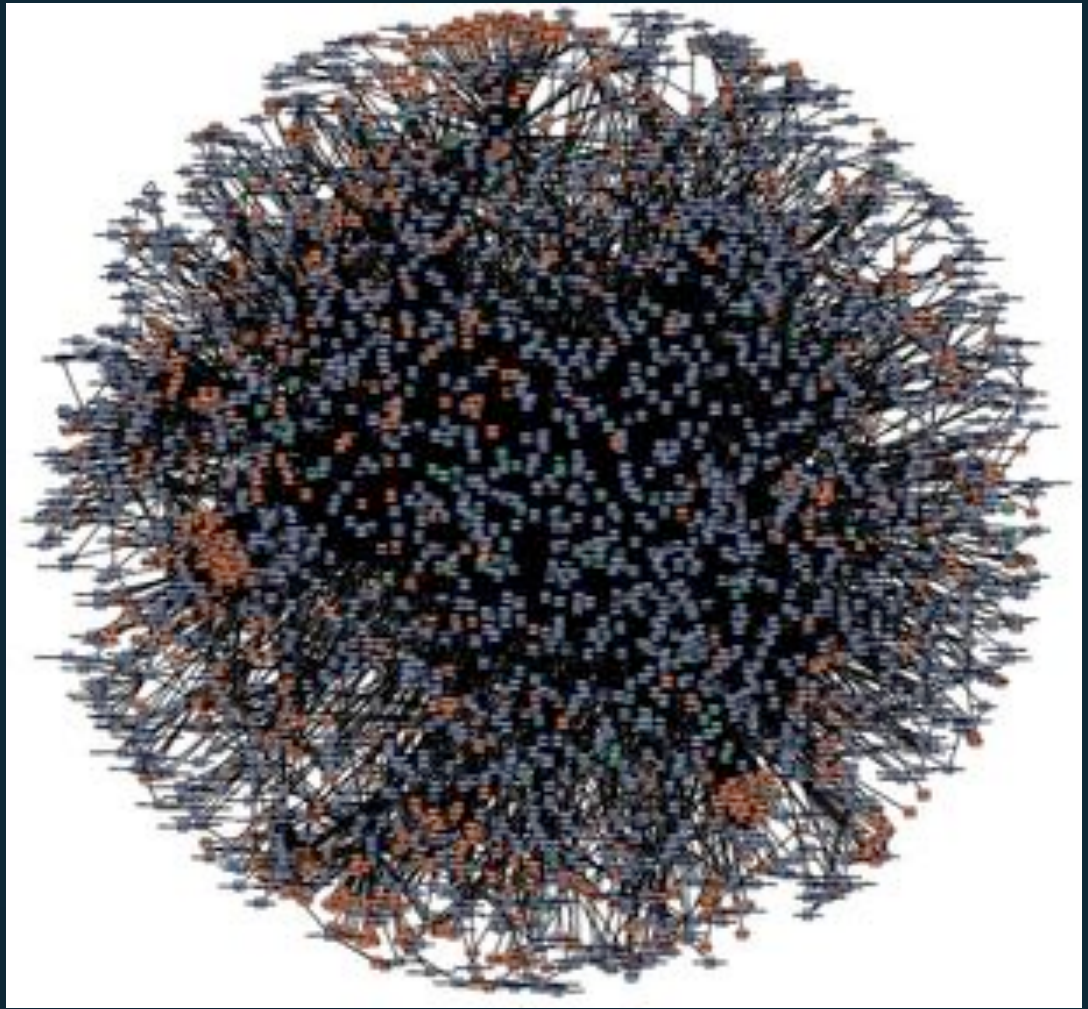
Two-pizza teams

Full ownership

Full accountability

Aligned incentives

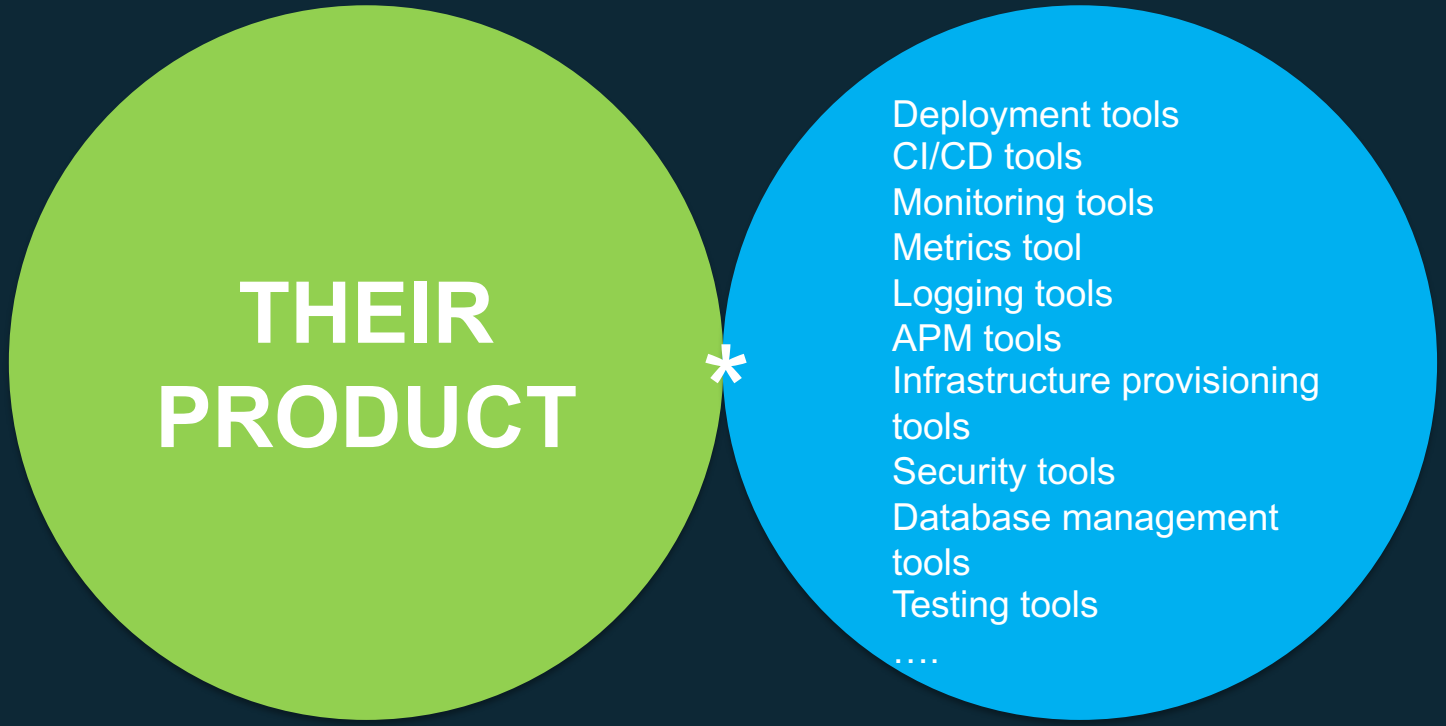
“DevOps”



Amazon S3 at launch:

Amazon S3 today:

2 Pizza Team Responsibility Venn Diagram



Responsible for

Not responsible for

*Unless their product belongs in the blue

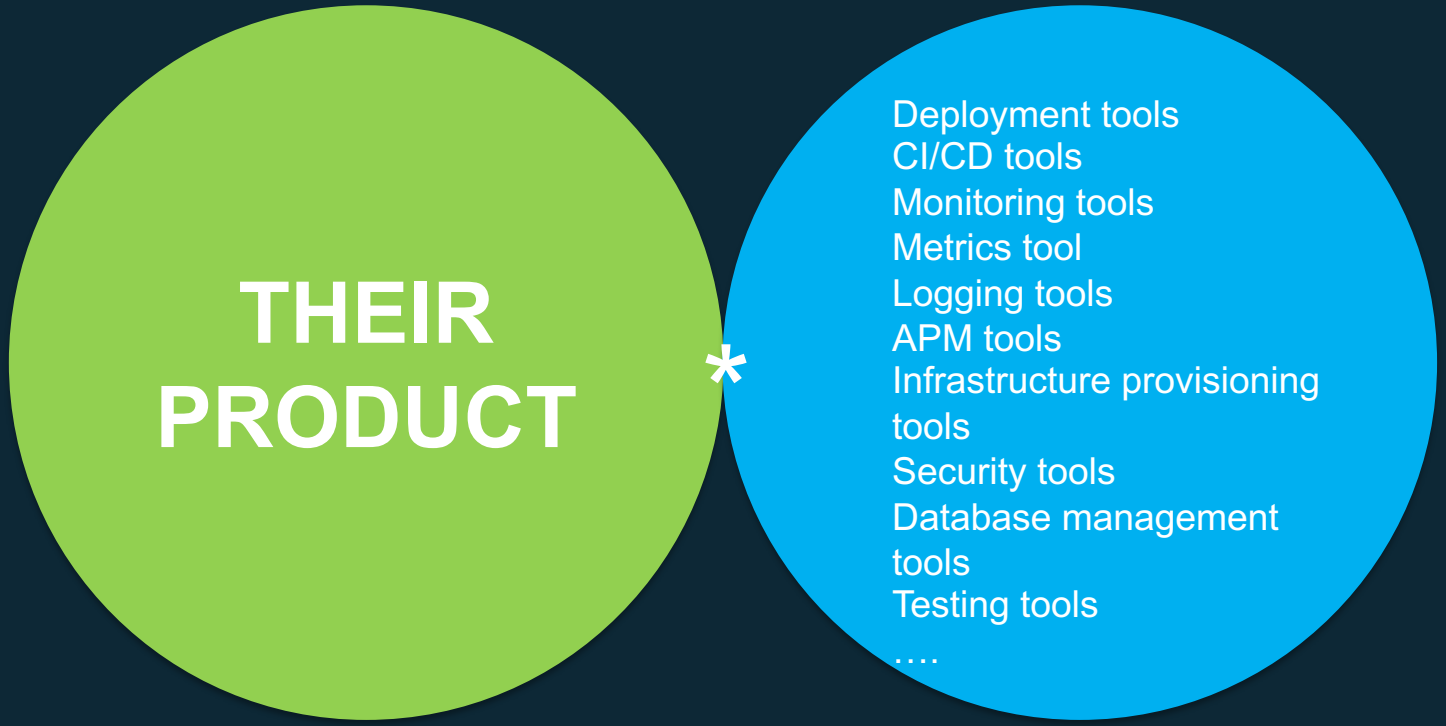
Self-service

Technology-agnostic

Encourage best practices

Single-purpose services

2 Pizza Team Responsibility Venn Diagram



Responsible for

Not responsible for

*Unless their product belongs in the blue

Deployment service

No downtime deployments

Health checking

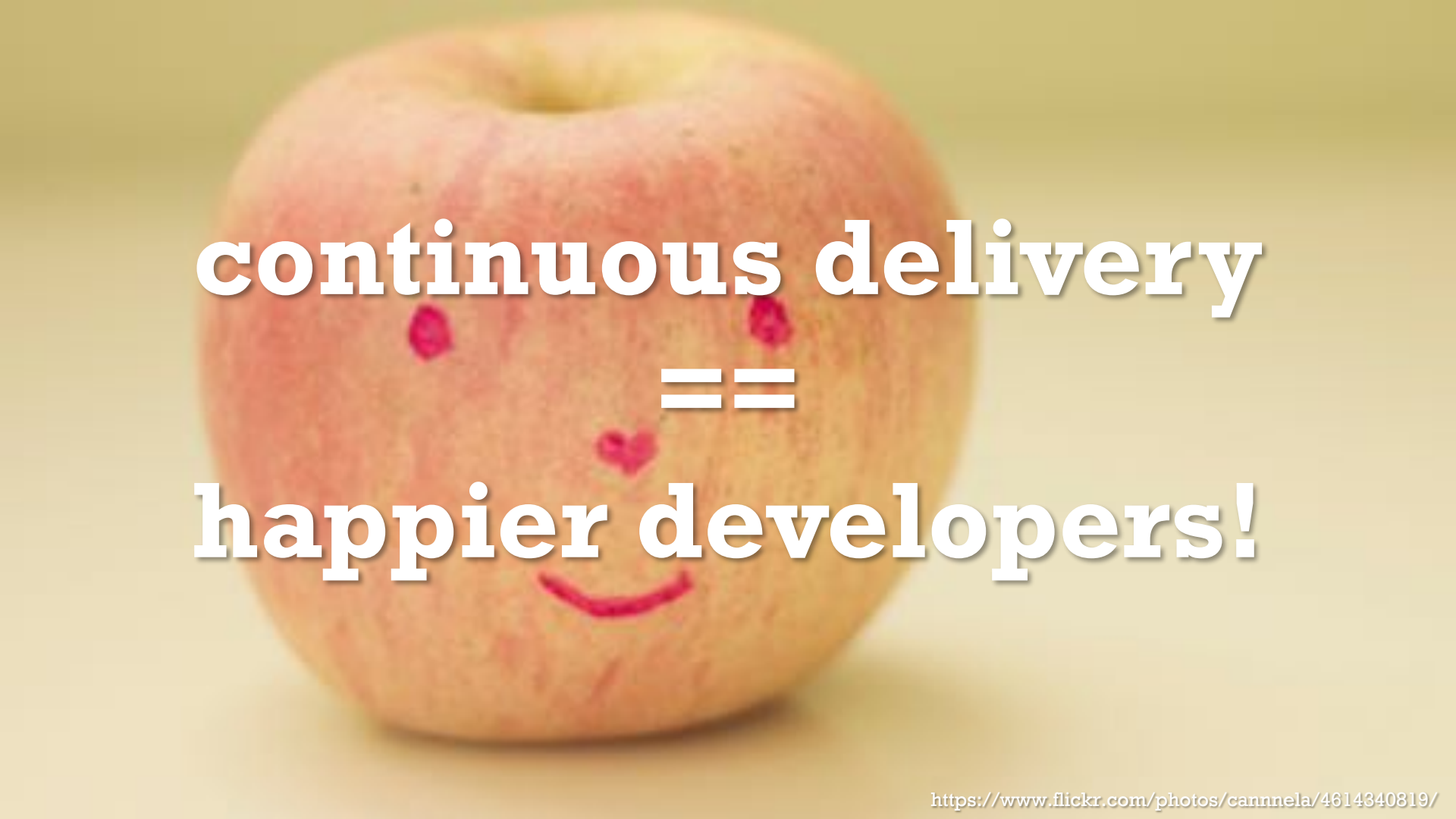
Versioned artifacts and rollbacks

Automated actions and transitions; from check-in to production

Development benefits:

- Faster
- Safer
- Consistent & Standardized
- Visualization of the process

aws


continuous delivery
==
happier developers!
https://www.flickr.com/photos/cannnela/4614340819/

Thousands of teams

× Microservice architecture

× Continuous delivery

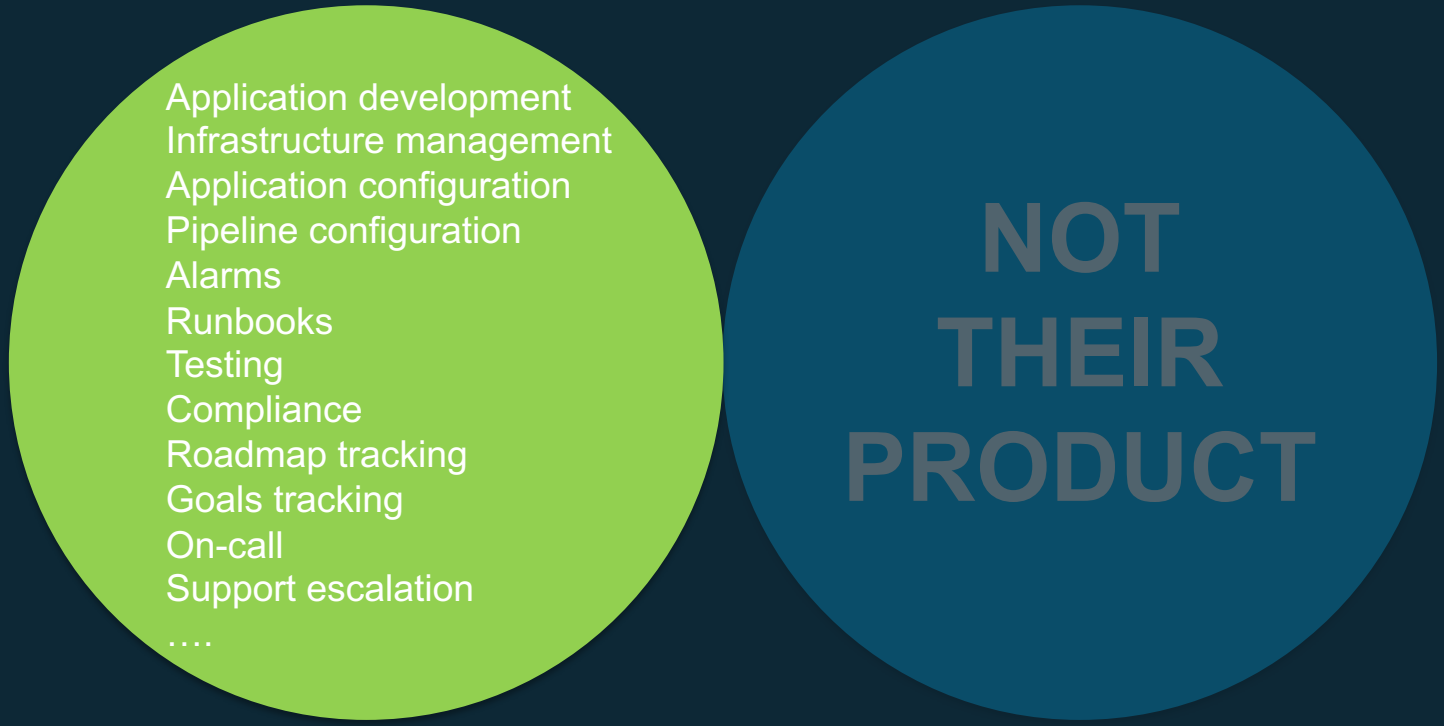
× Multiple environments

> 60 million deployments a year*

*2016 number

aws

2 Pizza Team Responsibility Venn Diagram



Responsible for

Not responsible for

*Unless their product belongs in the blue

Quick poll!

How many of you run in 7 or fewer AWS Regions?

How many of you run in 5 or fewer AWS Regions?

How many of you run in just 2 AWS Regions?

How many of you run in just 1 AWS Region?

aws

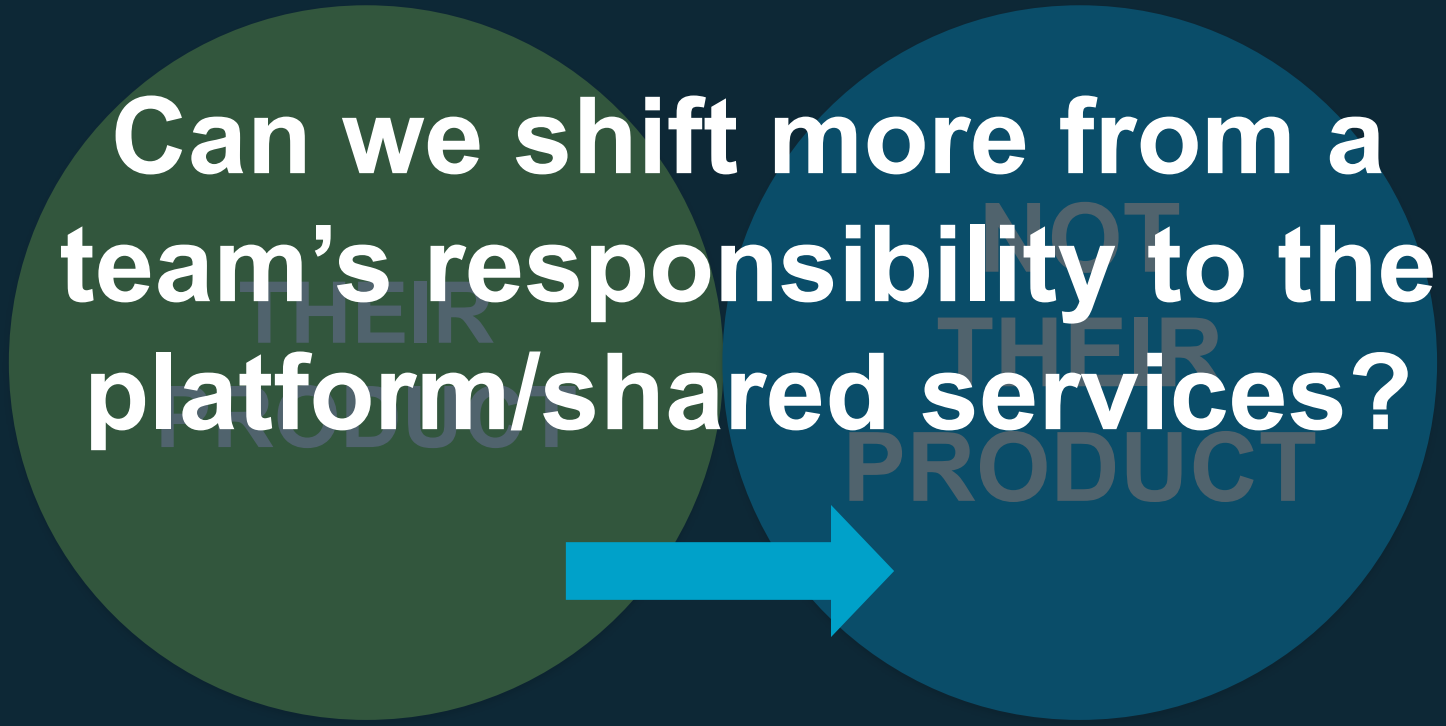
Global Infrastructure

https://aws.amazon.com/about-aws/global-infrastructure/

Determining the right balance

The more time spent on operational tasks, the less time spent on development tasks

2 Pizza Team Responsibility Venn Diagram



Can we shift more from a team's responsibility to the platform/shared services?

Responsible for

Not responsible for

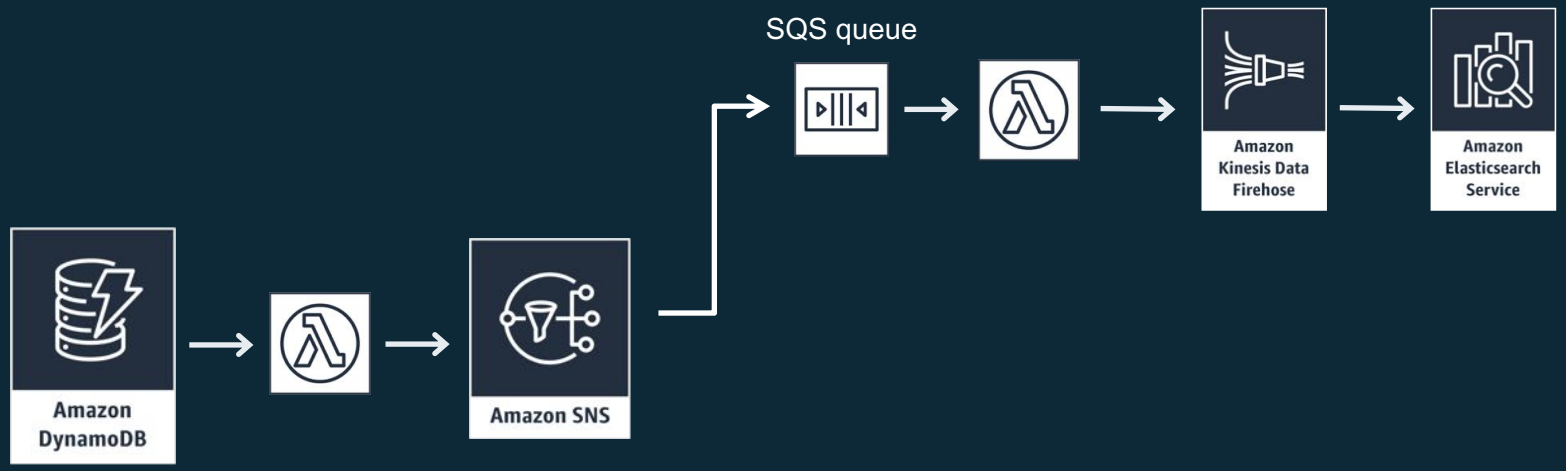
So that brings us back to…

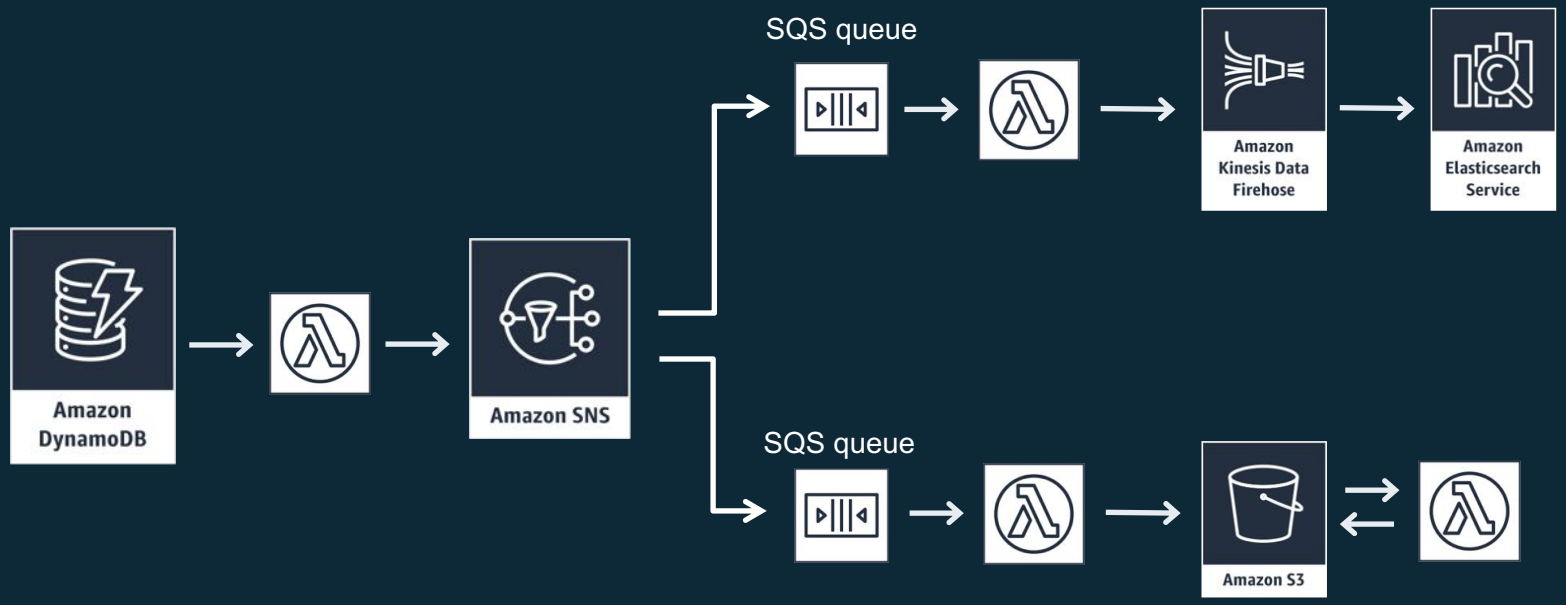
OK, let’s get real

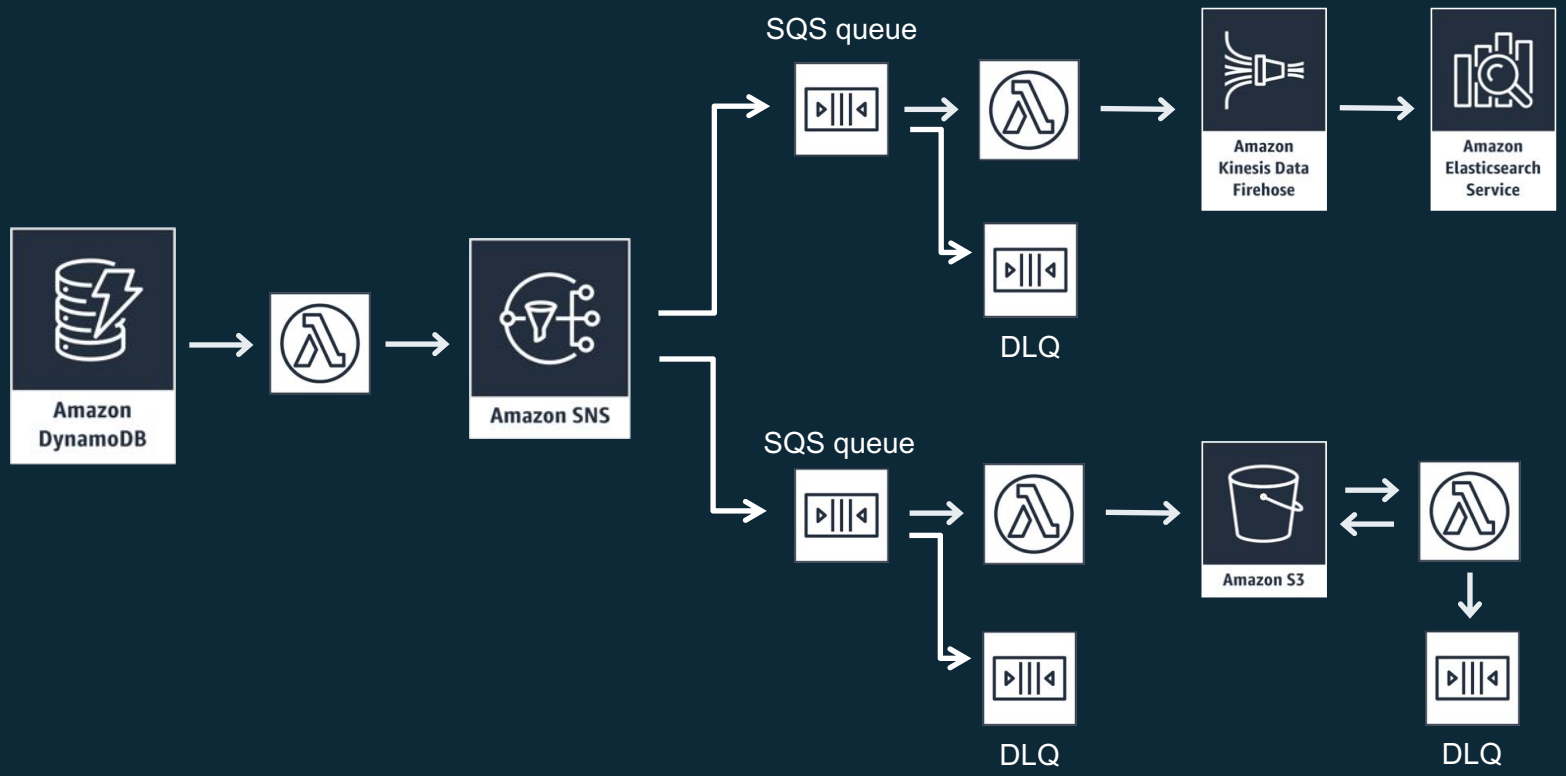
Photo by Clem Onojeghuo on Unsplash

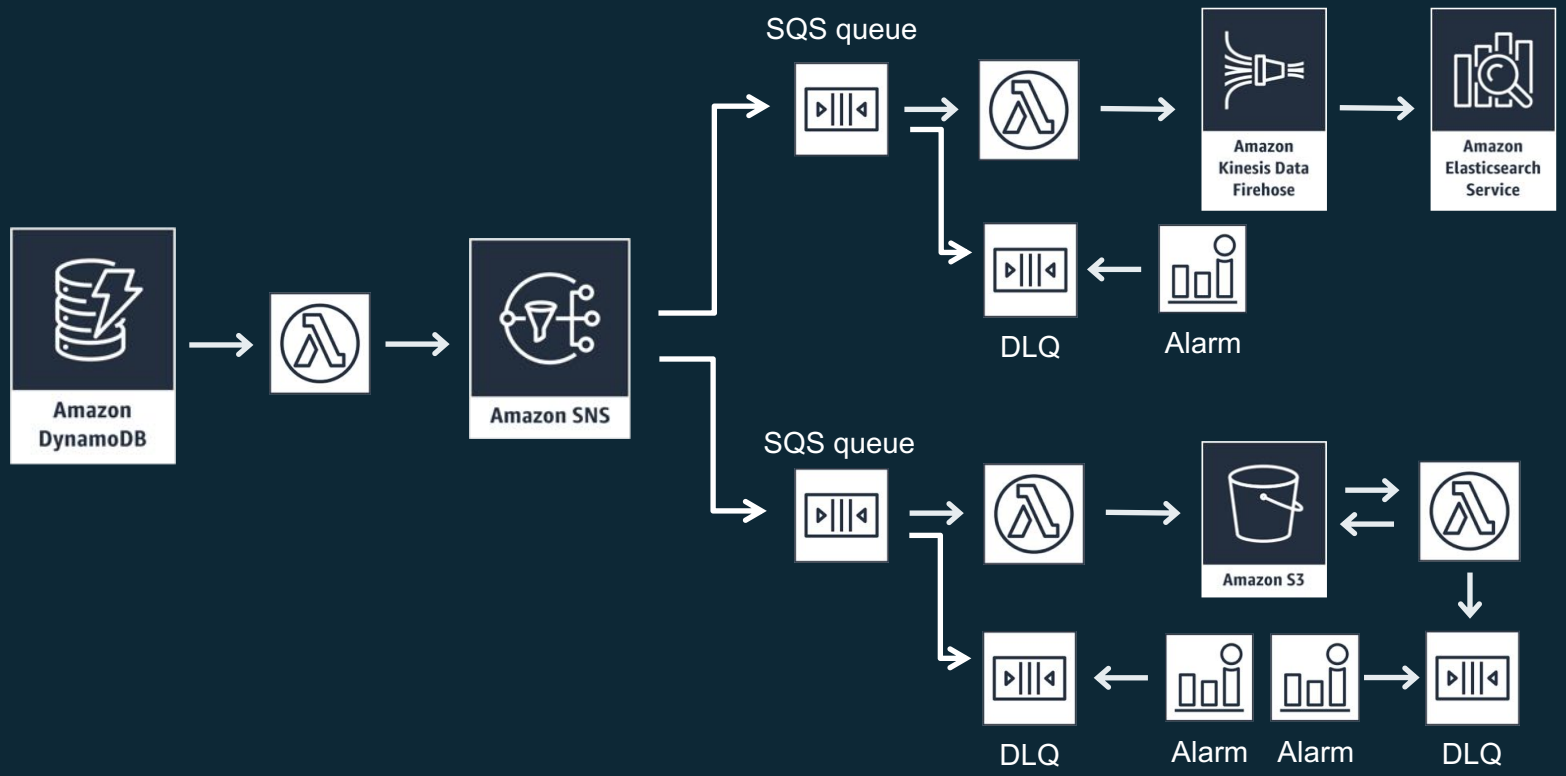
From the earliest days of AWS Lambda we saw lots of repeatable patterns (internally and externally)

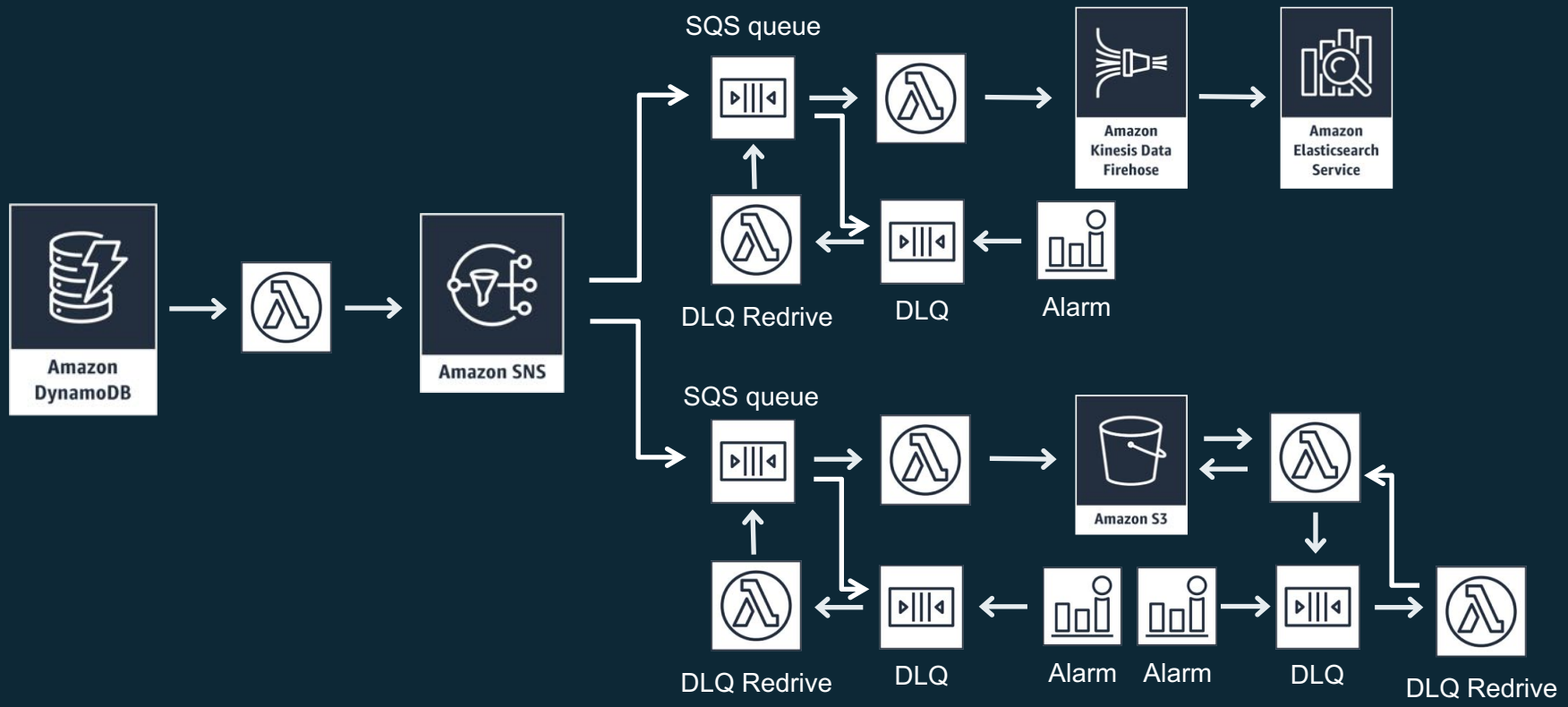
Photo by Kristian Strand on Unsplash

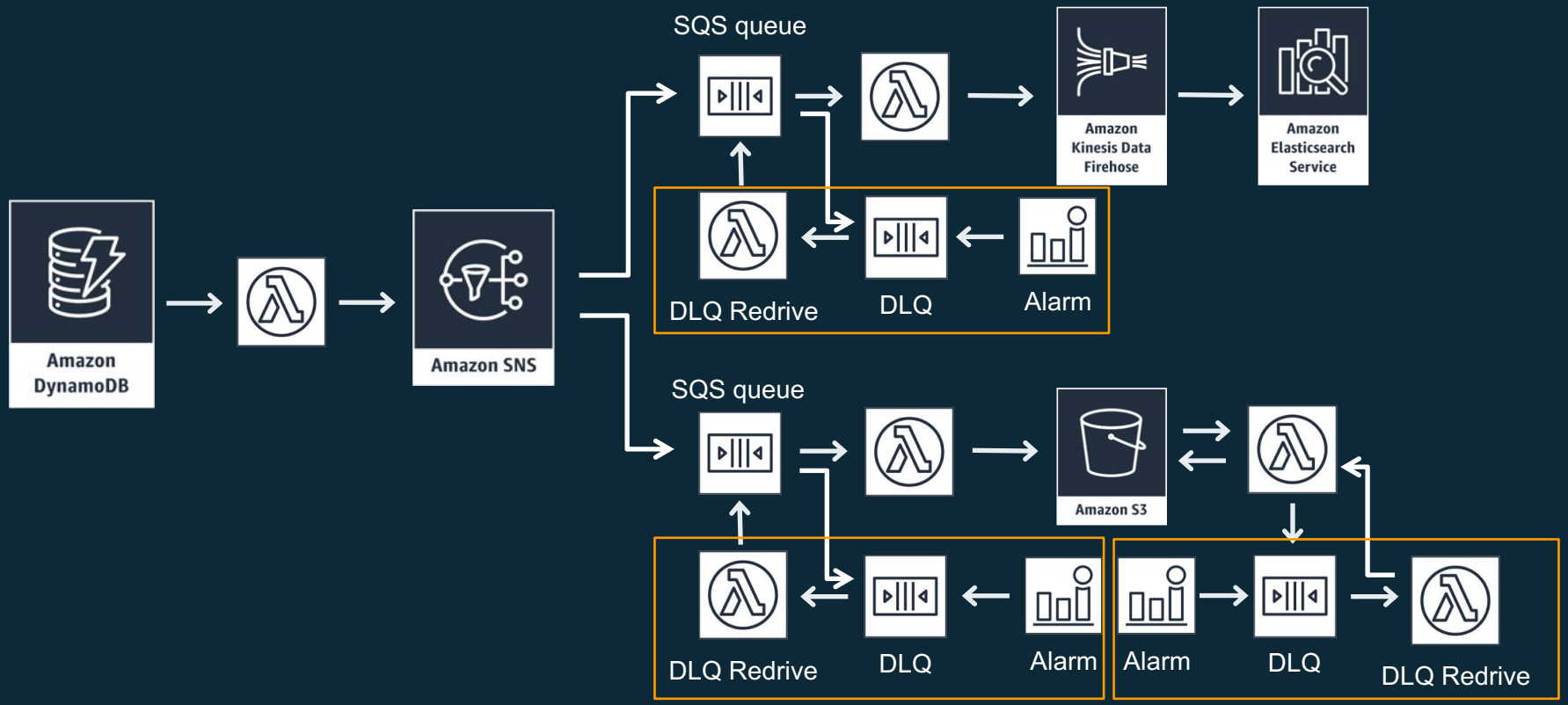


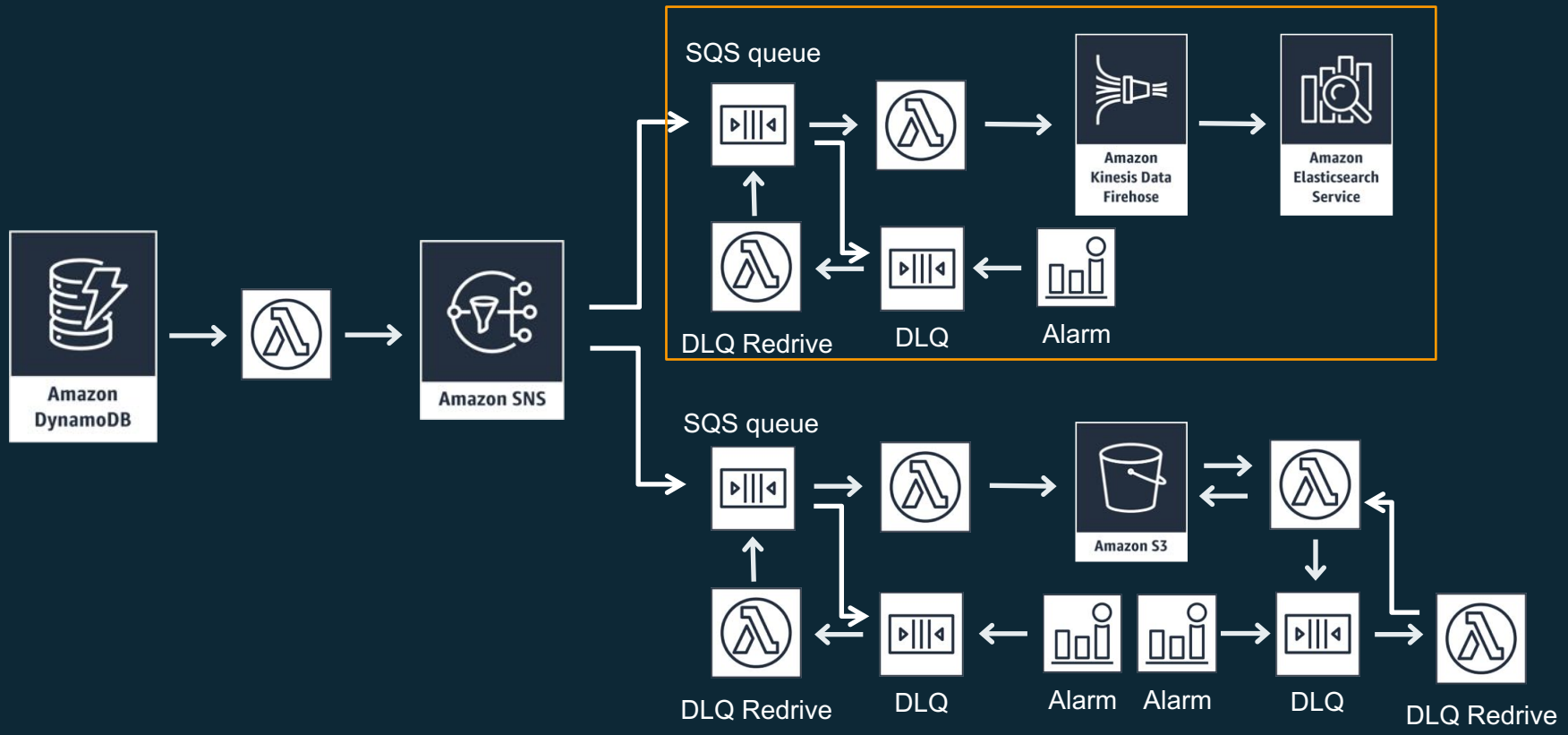












aws

Home > Applications

AWS Serverless Application Repository

Discover, deploy, and publish serverless applications

Applications (61)

Sort by Best Match

API

1 2 3 4 5 6

Voice-Lexicon-API

This app powers voicelexicon.com where you can see it in action. It accepts data from HTML form and generates MP3 so you can practice foreign language whenever you want.

Pavel Kral 11 deployments

Salesforce-API-Access-Manager-Monitor-Logger

A simple API access manager built on AWS lambda to provide multi tiered access to salesforce services with a single API user. Please read more here: https://github.com/manjit5190/Salesforce-API-Access-Manager-Monitor-Logger/blob/master/README.md

salesforce-api-access-manager

MIS 0 deployments

MEI

Japanese Zipcode API

zipcode api classmethod

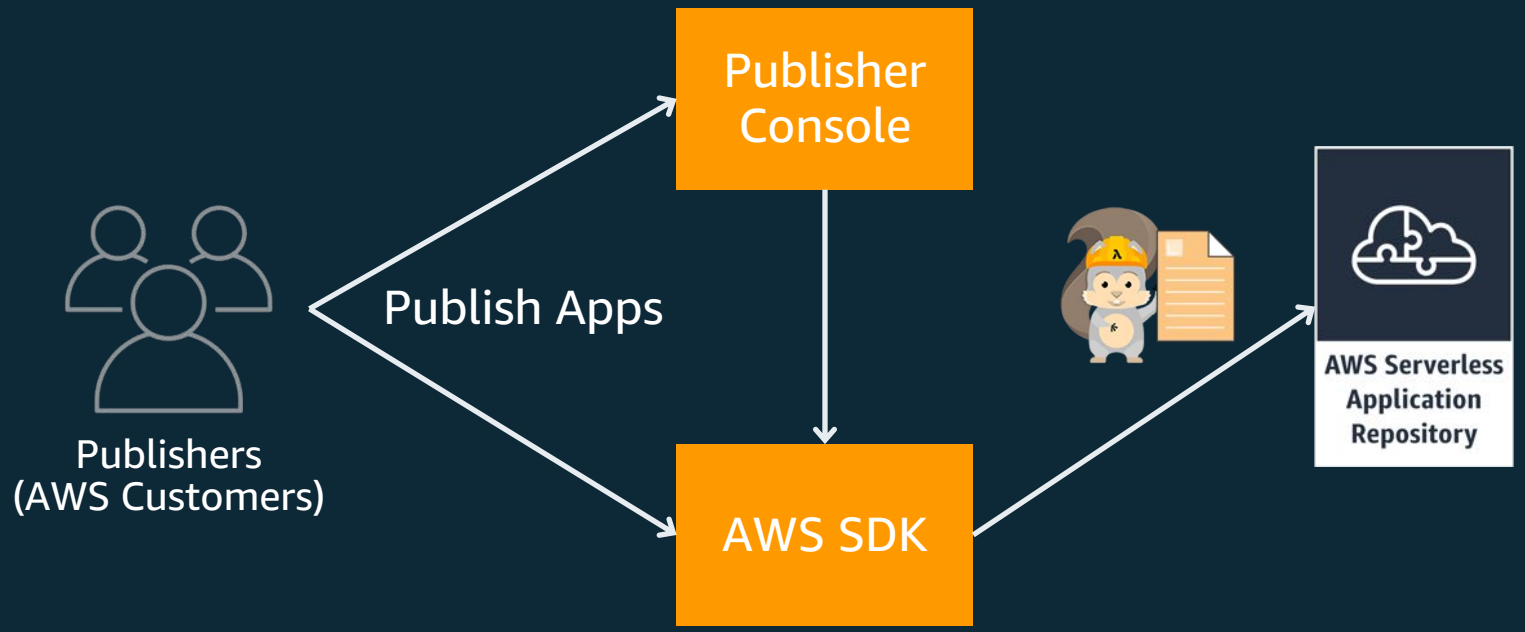
Classmethod Inc 3 deployments

microservice-http-endpoint-python

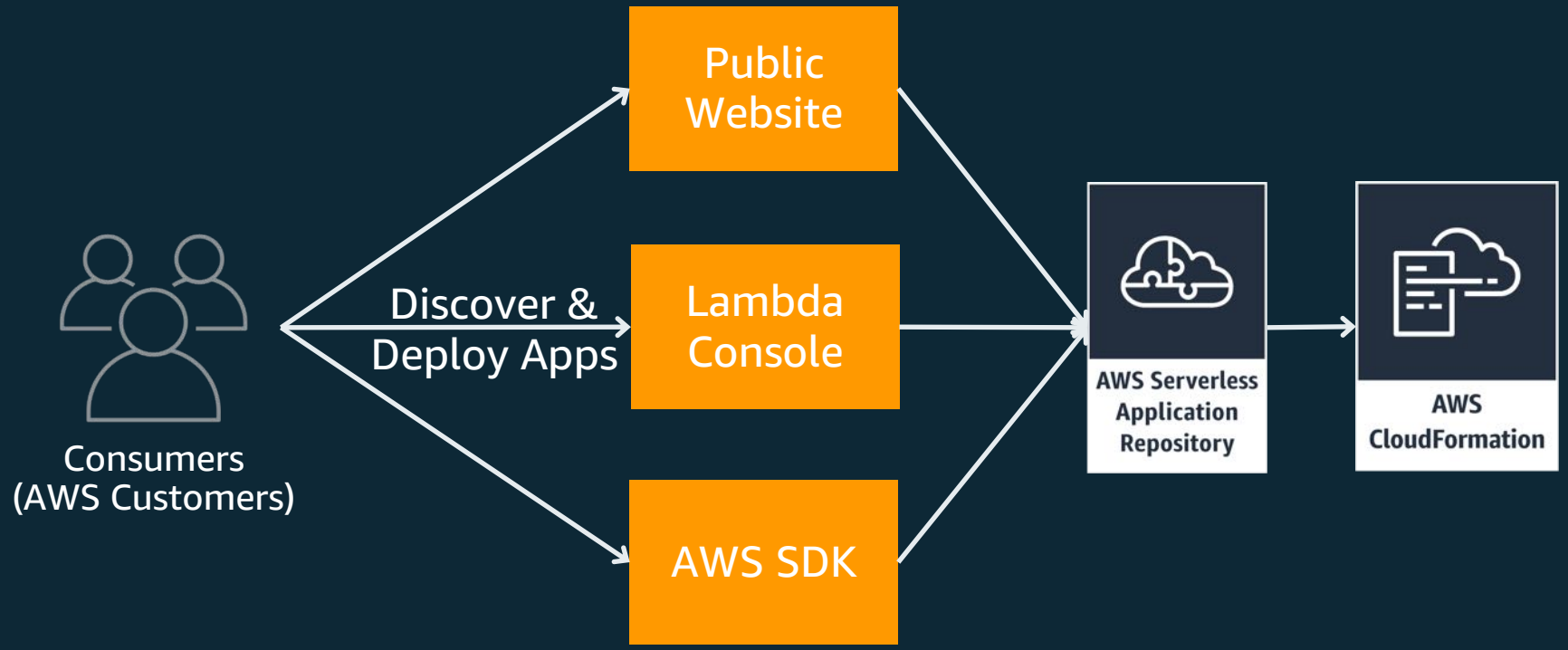
microservice-http-endpoint-python3

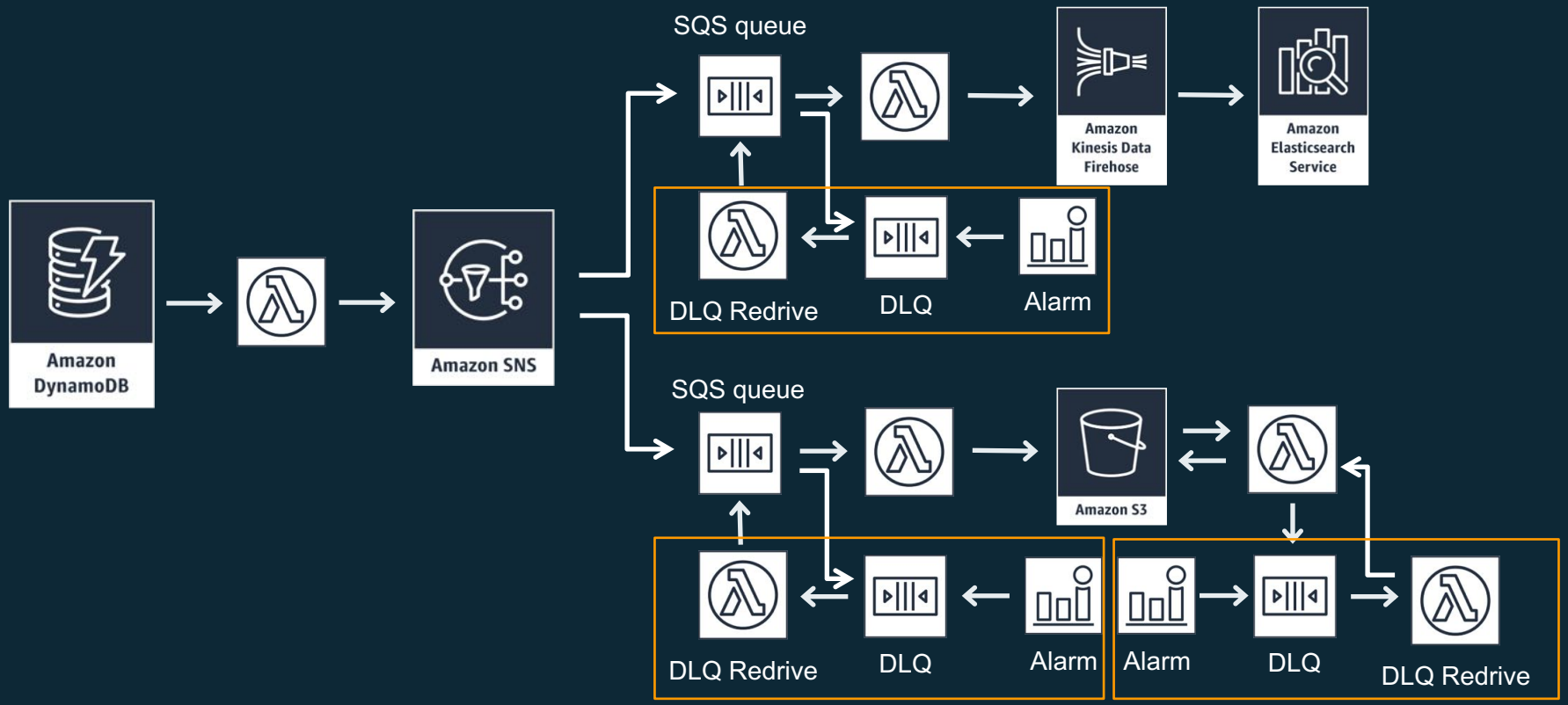
microservice-http-endpoint

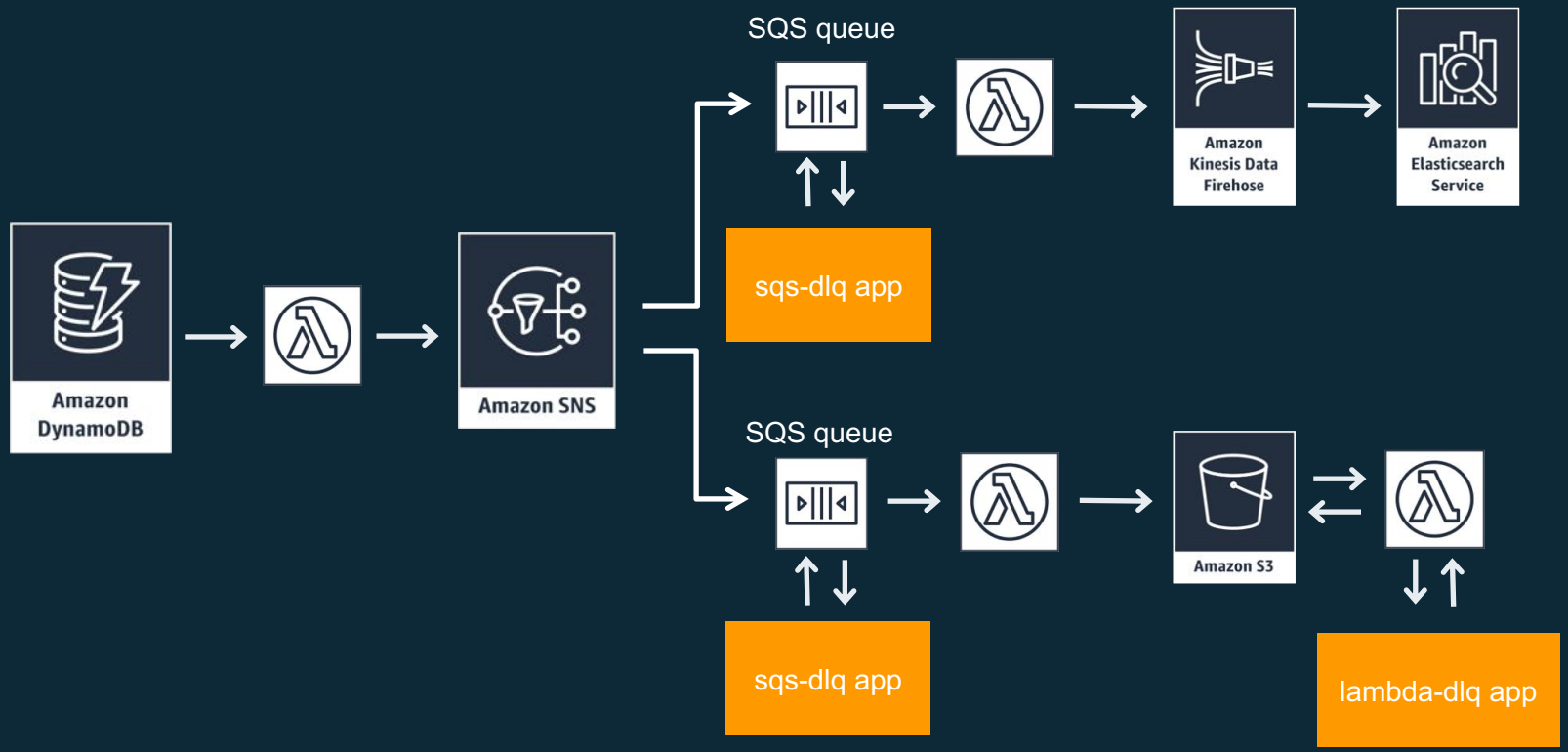
Publishing Applications

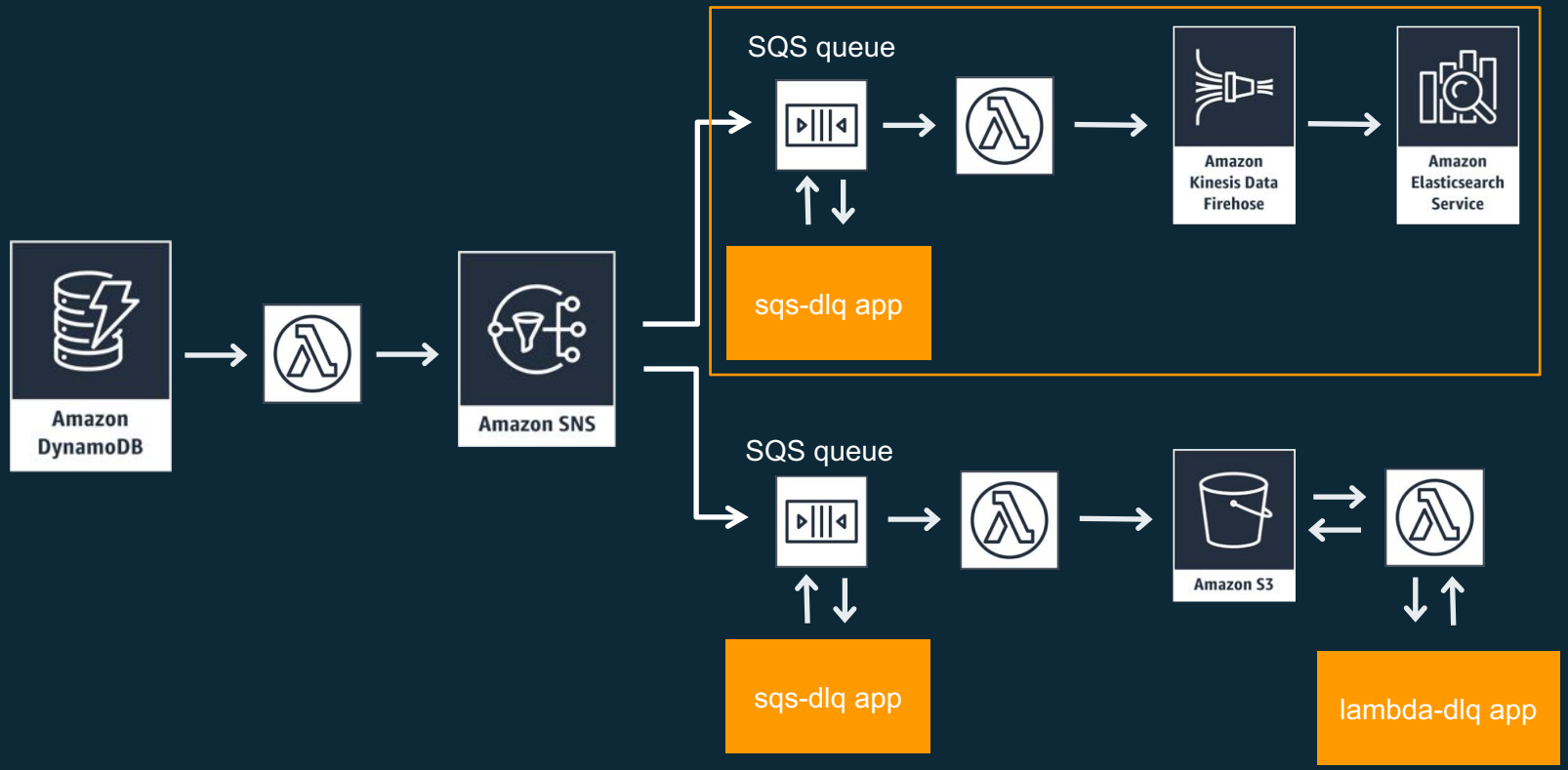


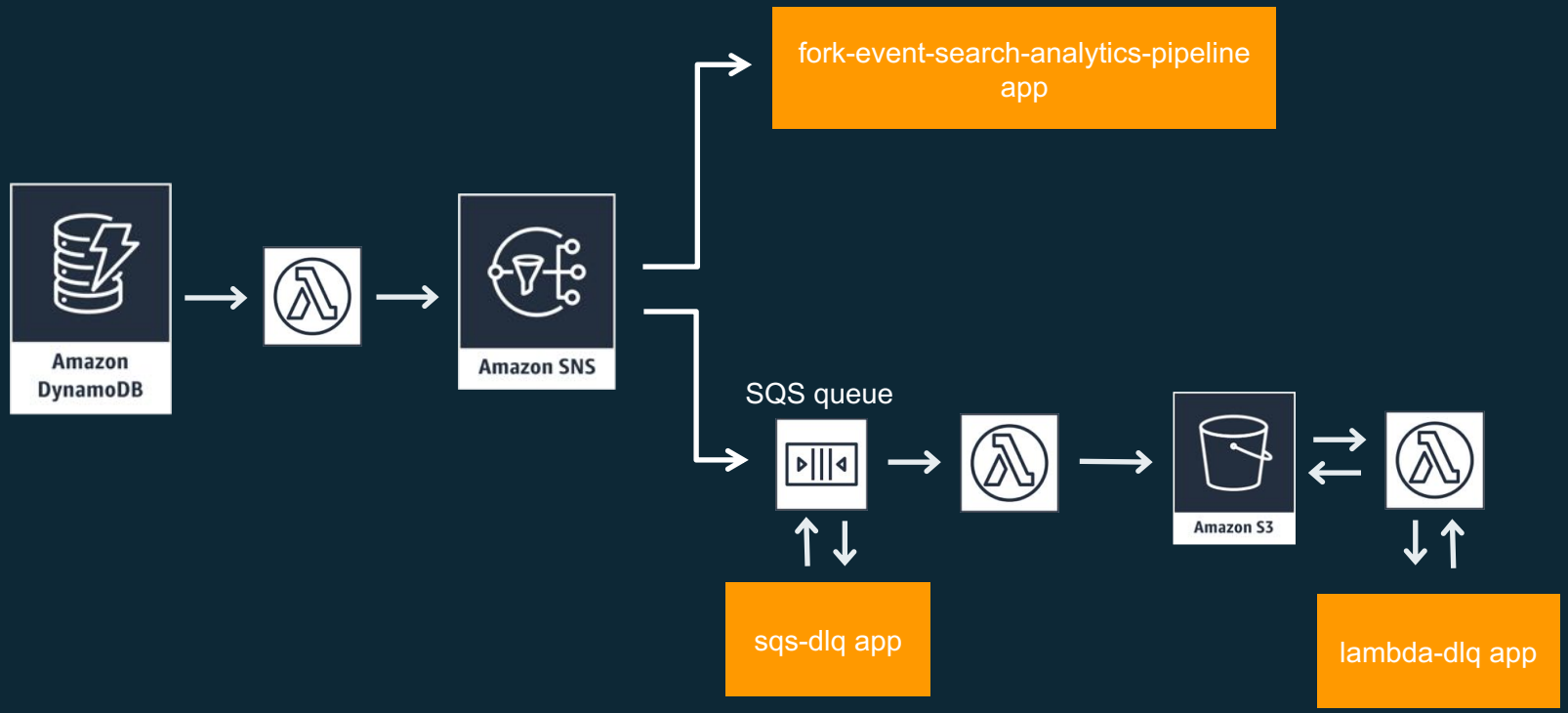
Consuming Applications











AWS Serverless Application Repository API

class **ServerlessApplicationRepository.Client**

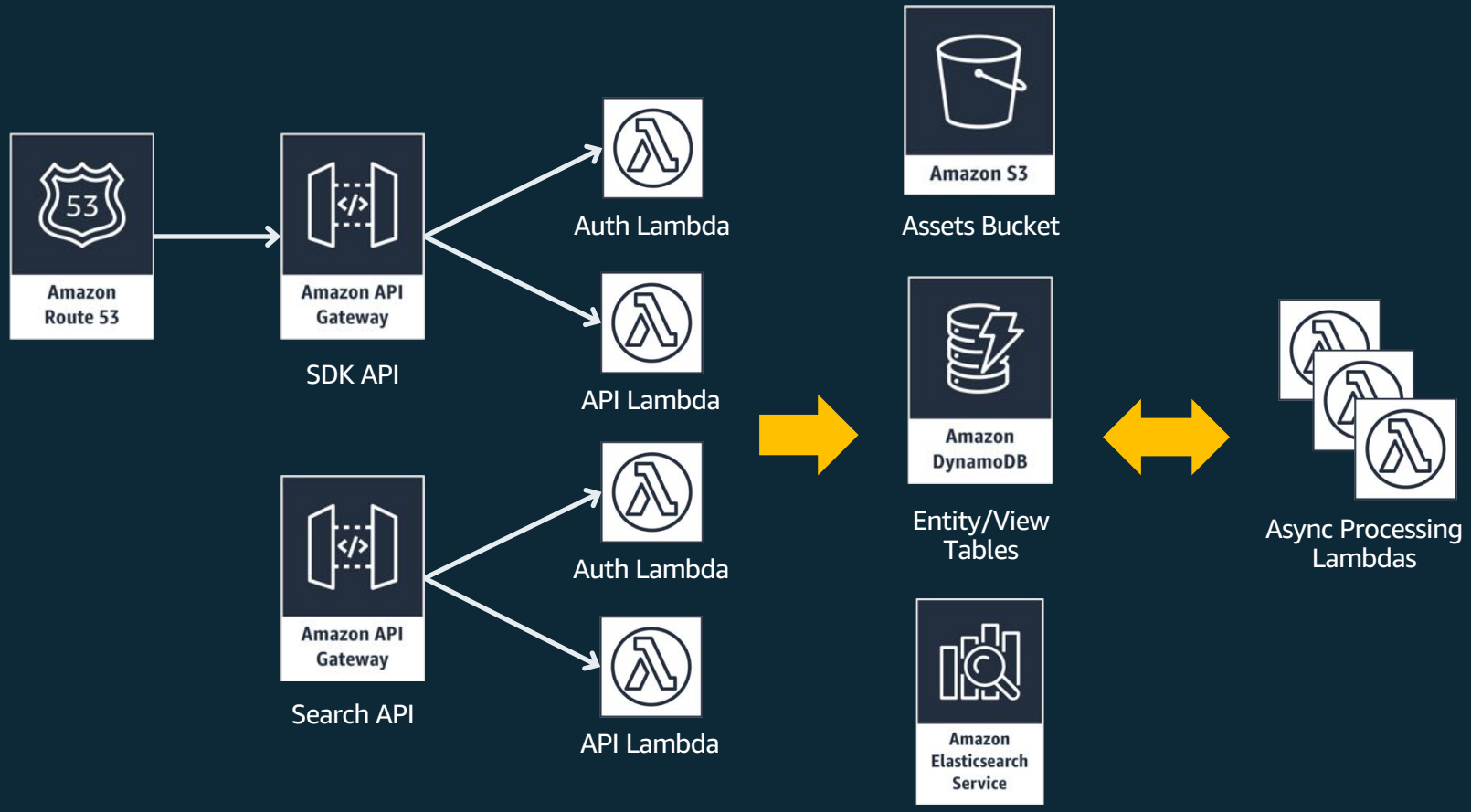
A low-level client representing AWSServerlessApplicationRepository:

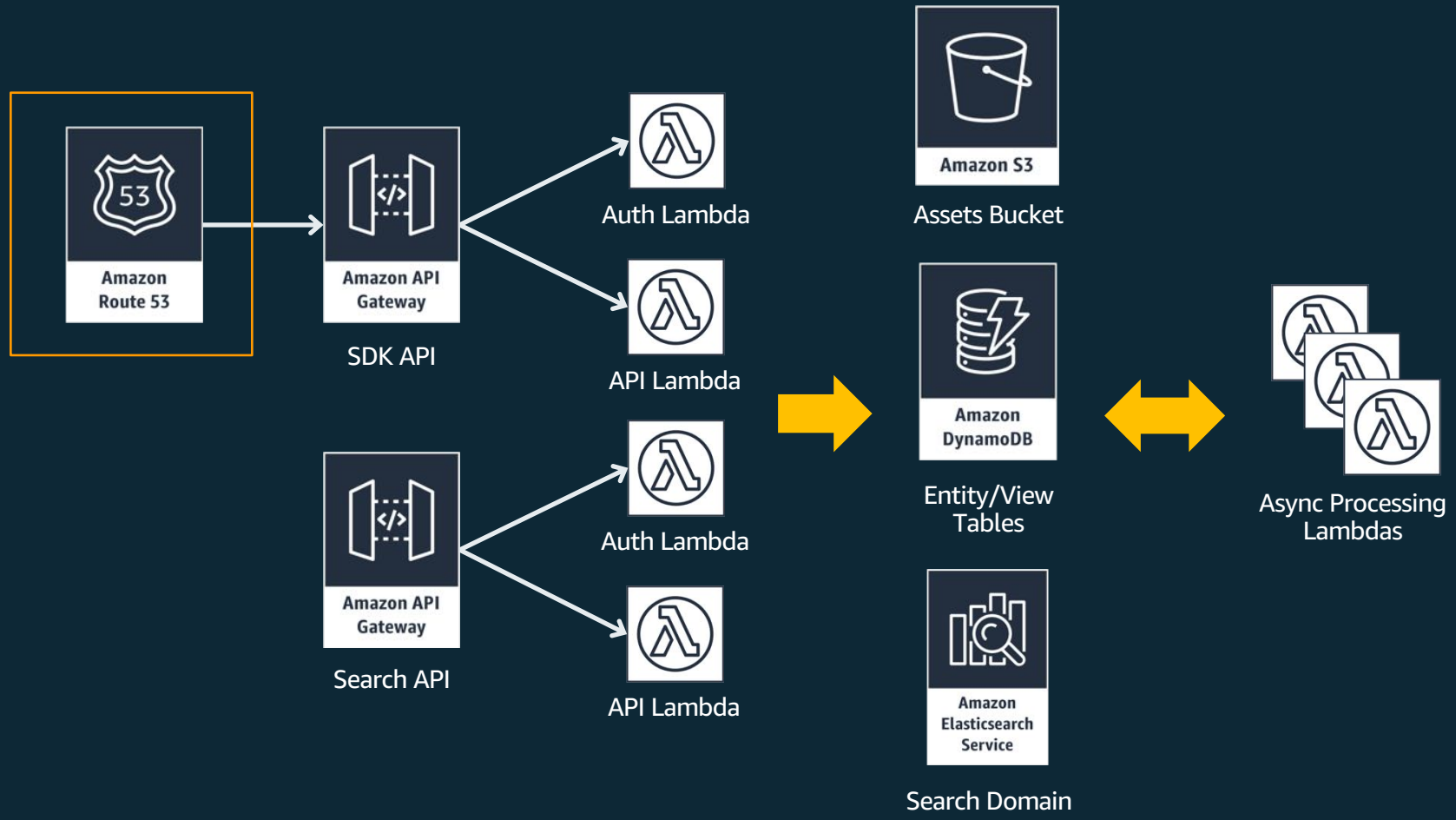
```
import boto3

client = boto3.client('serverlessrepo')
```

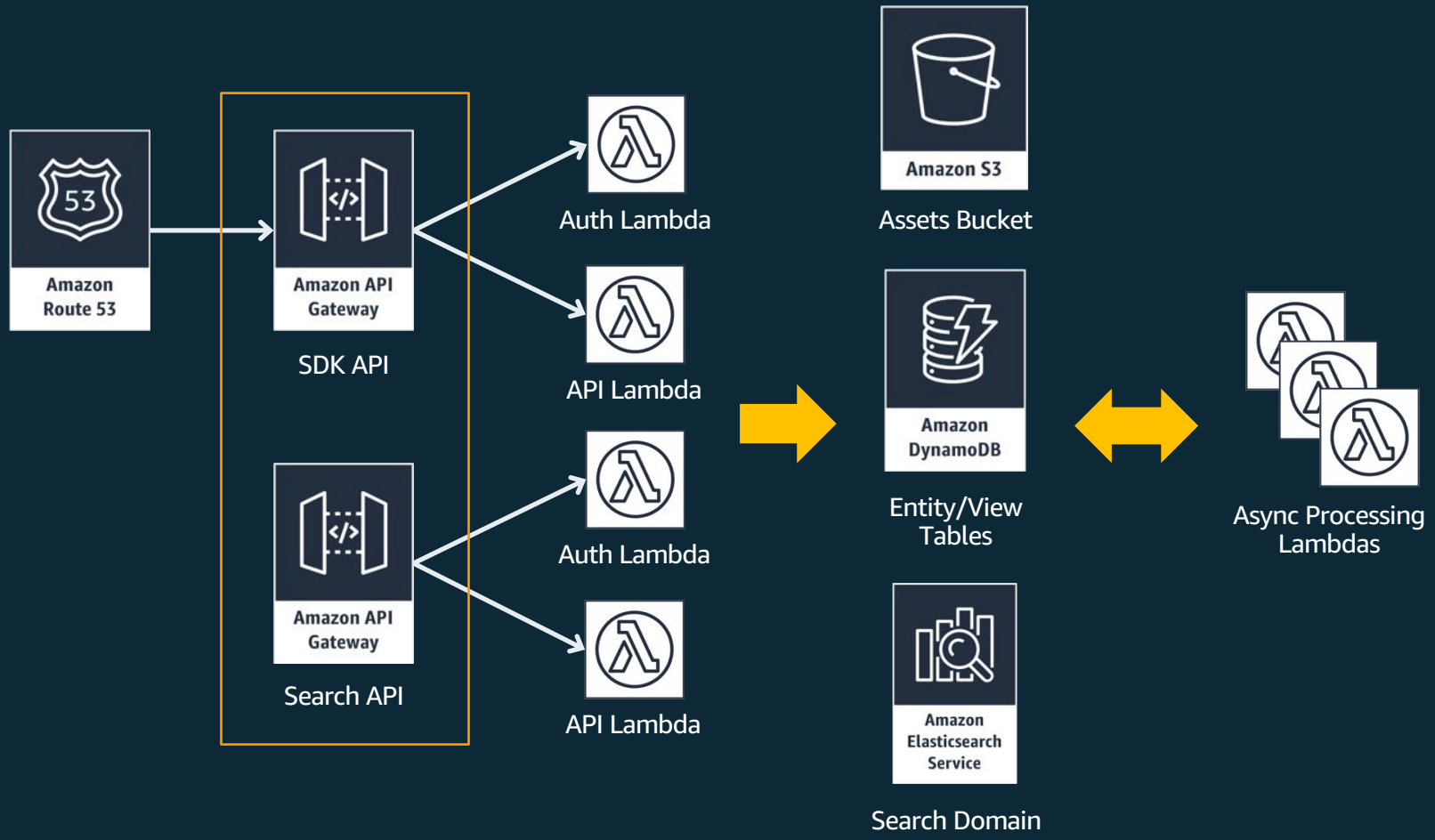
These are the available methods:

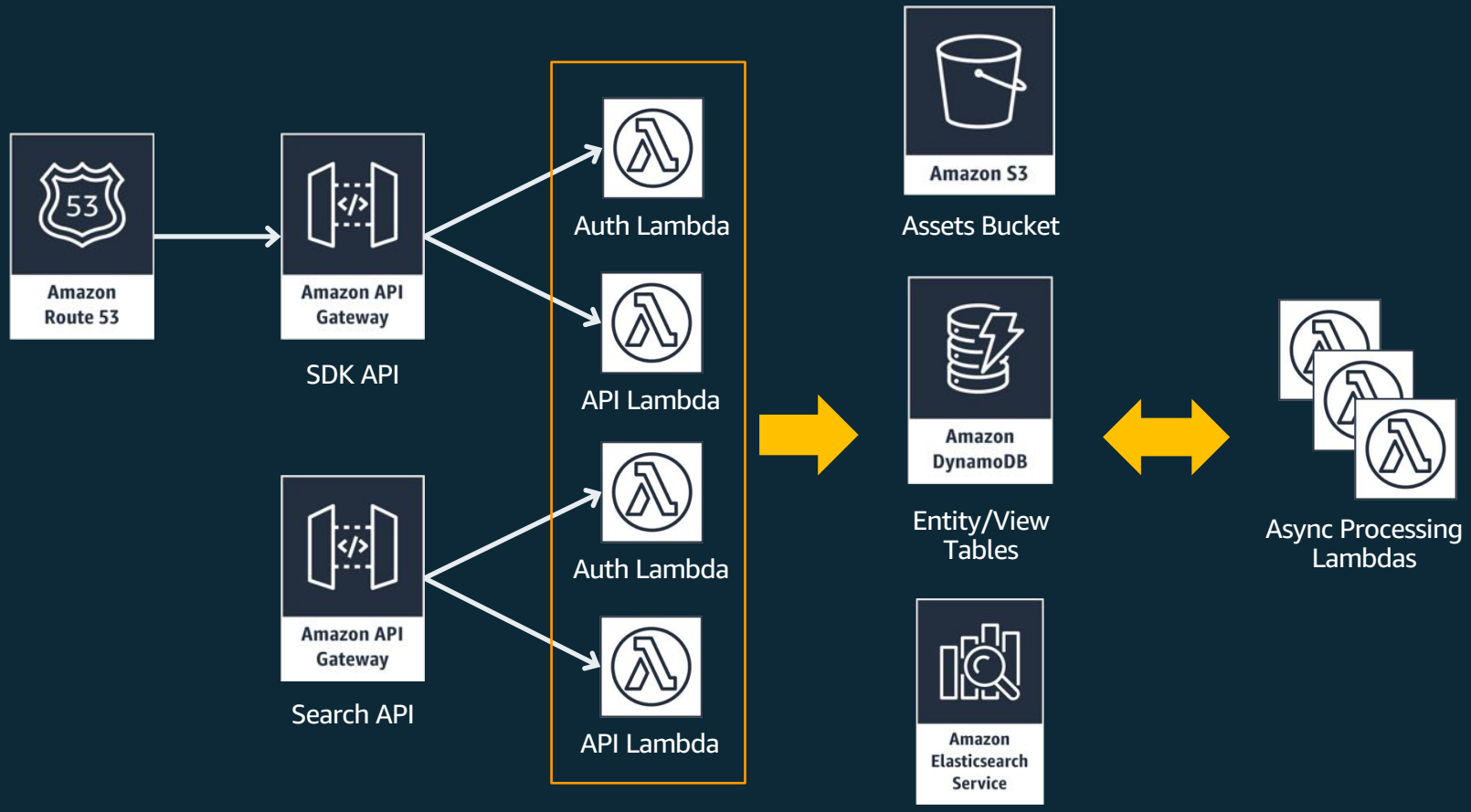
- can_paginate()
- create_application()
- create_application_version()
- create_cloud_formation_change_set()
- create_cloud_formation_template()
- delete_application()
- generate_presigned_url()
- get_application()
- get_application_policy()
- get_cloud_formation_template()
- get_paginator()
- get_waiter()
- list_application_dependencies()
- list_application_versions()
- list_applications()
- put_application_policy()
- update_application()

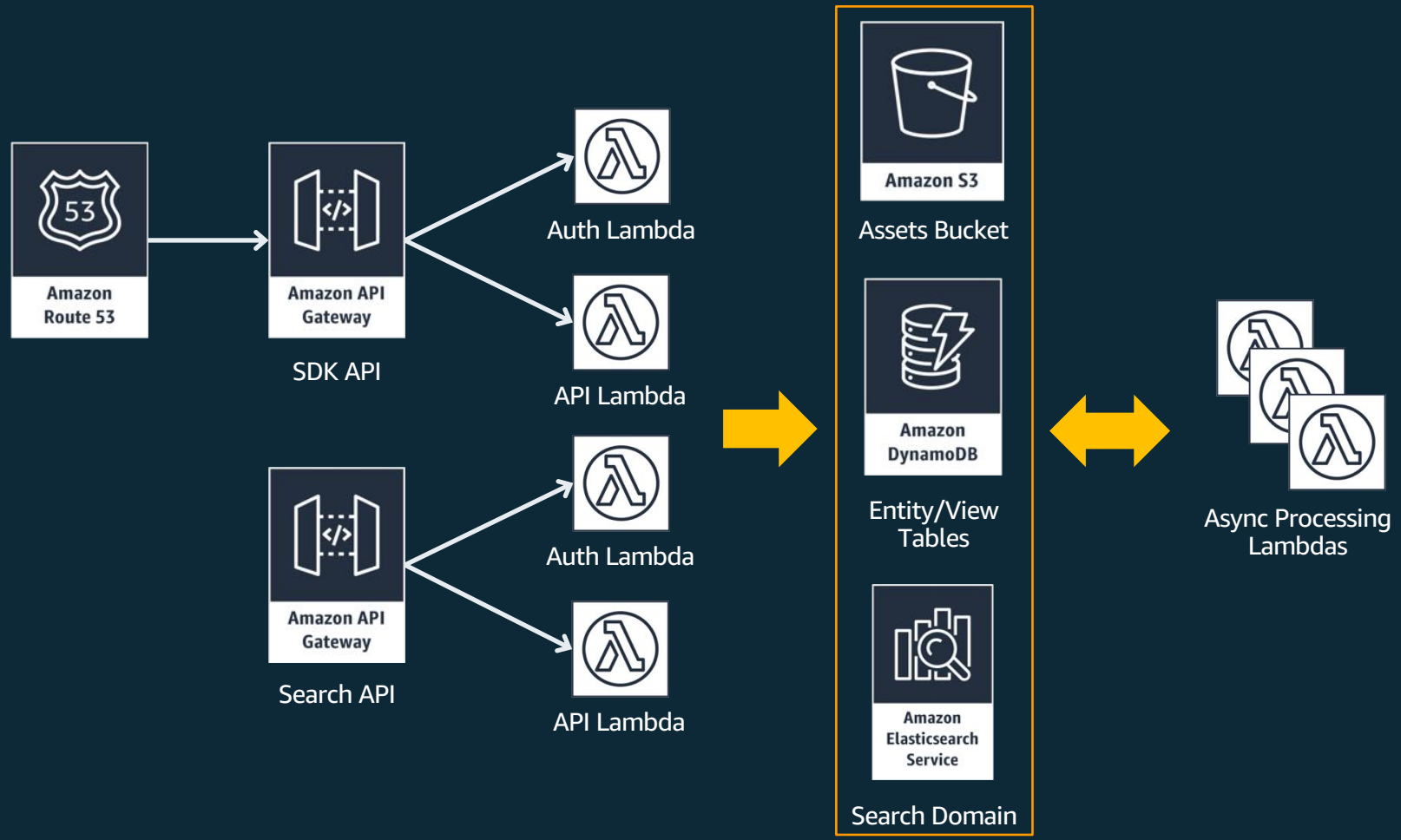
https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/serverlessrepo.html

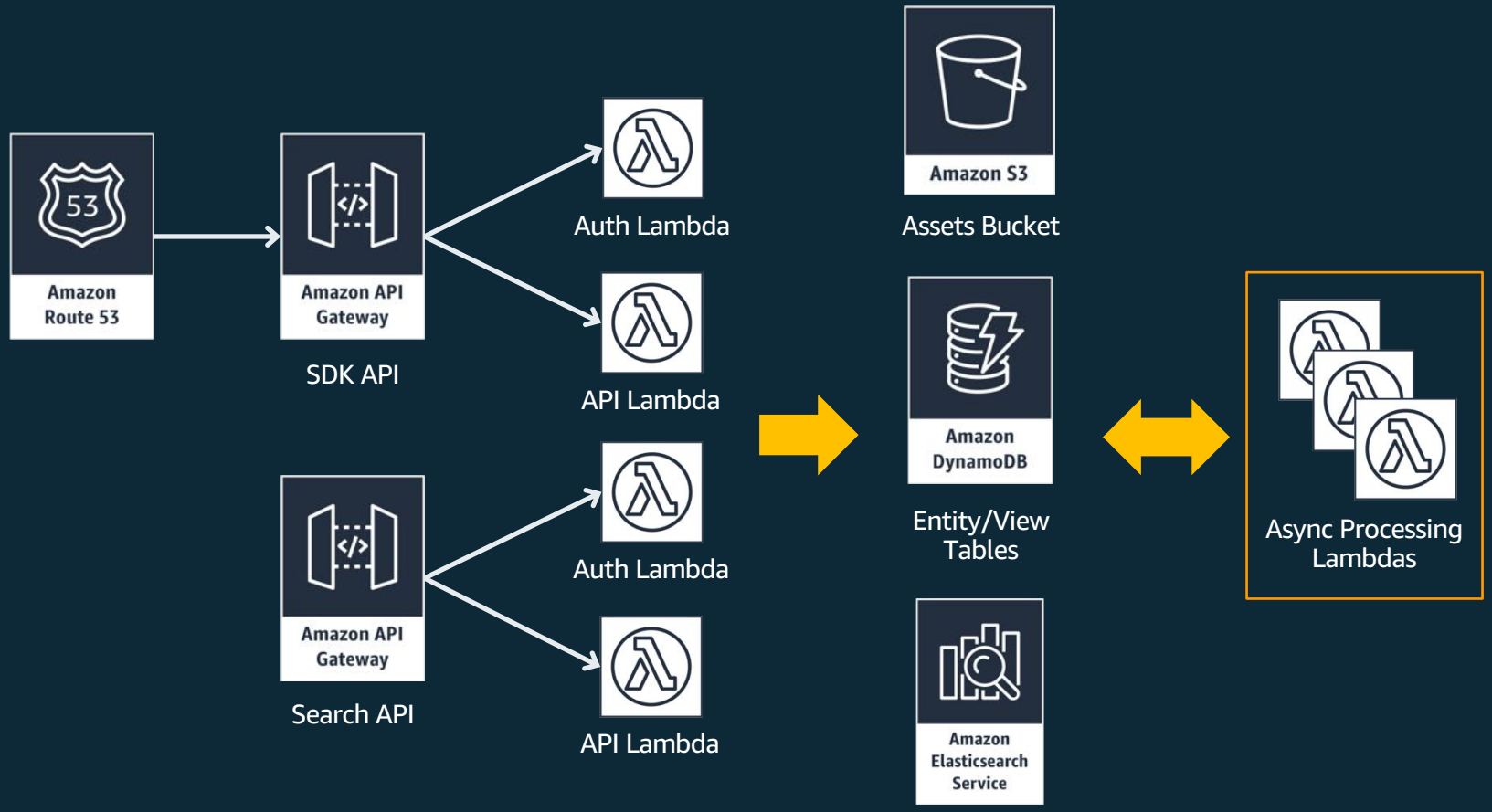












Not seen

- Async processing Lambdas interfacing with Amazon SNS/Amazon SQS
- A few other internal services
- Alpha/beta/gamma
- Multiple regions

Not seen

- Async processing Lambdas interfacing with Amazon SNS/Amazon SQS
- A few other internal services
- Alpha/beta/gamma
- Multiple regions

- A single Amazon EC2 instance per region, to test that the service is working properly

  BEST PRACTICE ALERT: Don’t test from the managed service to managed service to be tested

Did building serverless-ly mean a massive shift in tools/practices?

Photo by Carlos Irineu da Costa on Unsplash

Nope.

Building with Serverless services at AWS

- Using Amazon CloudFormation/AWS SAM to manage infrastructure
- Leveraging existing CI/CD tools that have been tweaked to support tools like SAM
- Difference in telemetry data made up for with more logging and usage of tools like AWS X-Ray
- Same languages as always: Java, Python, Node.js
- Little to no changes in development/testing practices:
    - peer review
    - test in “cloud”
    - linting, syntax checking, unit/service testing
    - canary deploy out slowly & rollback immediately if things go boom

aws

“Tell me your biggest pain points right now”

Photo by Zhen Hu on Unsplash

Tell me your biggest pain points right now:

Limits

Scoping the size of a service

Proper “stack” boundaries

Improving alarms/monitoring

Improving testing (always)

Tell me your biggest pain points right now:

Limits

Scoping the size of a service

Proper "stack" boundaries

Improving alarms/monitoring

Improving testing (always)

AWS Service teams are just like you!

Tell me your biggest pain points right now:

Limits

Scoping the size of a service

Proper "stack" boundaries

Improving alarms/monitoring

Improving testing (always)

AWS Service teams are just like you!

Except…….

What you can do to be like AWS

- Organizational structure aligns with microservice based architecture which aligns well with serverless
- Standardized, yet flexible tools made supporting serverless essentially “snapping in” new capabilities
- CI/CD is a must
- Peer review is the one thing you aren’t doing but should
- Understanding when you need a REST API vs. using another invoke model
    - related: Sync vs. async

aws

What you can do to be like AWS

- If moving to #serverless feels like a giant shift for your org/team, move slow and purposefully
- If someone is telling you that you need all new tools to develop for serverless, they are wrong or selling you something
    - Potentially something that competes

What customers tell me:

Serverless benefits pay off vs. the time ramping up

Photo by Raphael Schaller on Unsplash

That’s kind of the idea…

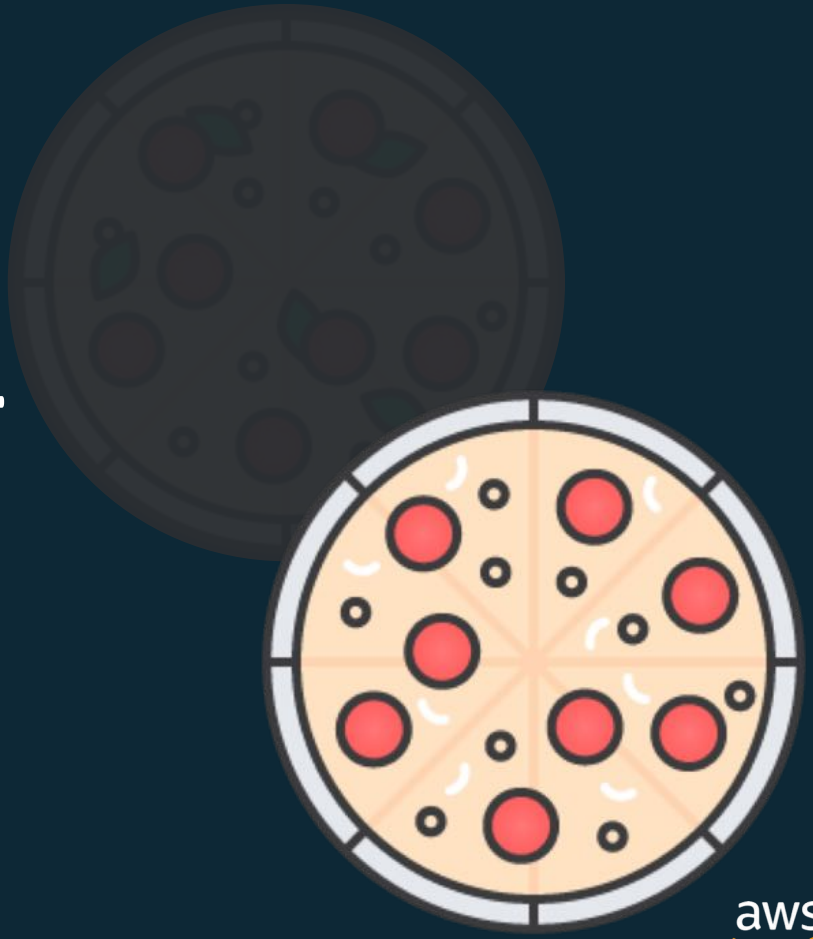
and it can lead to an exciting future


Going back to Amazon
Photo by Chris Munns

What happens if we reduce operational burden massively and reduce the amount of code we have to write?

aws

What if 2 pizzas is 1 too many?



?x more Thousands of teams

× Microservice architecture
× Continuous delivery
× Multiple environments

If fewer people are needed to build things, can we build more things & faster???

> XXX million deployments a year?

About me:

Chris Munns - munns@amazon.com, @chrismunns

- Principal Developer Advocate - Serverless
- New Yorker
- Previously:
  - AWS Business Development Manager – DevOps, July ’15 - Feb ‘17
  - AWS Solutions Architect Nov, 2011- Dec 2014
  - Formerly on operations teams @Etsy and @Meetup
  - Little time at a hedge fund, Xerox and a few other startups
- Rochester Institute of Technology: Applied Networking and Systems Administration ’05
- Internet infrastructure geek

aws.amazon.com/serverless

Menu | aws | Contact Sales | Products | Solutions | Pricing | More | English | My Account | Sign In to the Console

Serverless Computing | Overview | AWS Serverless Application Repository | Developer Tools | Resources | Partners

Serverless Computing and Applications

Build and run applications without thinking about servers

Find serverless applications

Serverless computing allows you to build and run applications and services without thinking about servers. Serverless applications don't require you to provision, scale, and manage any servers. You can build them for nearly any type of application or backend service, and everything required to run and scale your application with high availability is handled for you.

Building serverless applications means that your developers can focus on their core product instead of worrying about managing and operating servers or runtimes, either in the cloud or on-premises. This reduced overhead lets developers reclaim time and energy that can be spent on developing great products which scale and that are reliable.

Chris Munns
munns@amazon.com
@chrismunns

https://www.flickr.com/photos/theredproject/3302110152/

https://secure.flickr.com/photos/dullhunk/202872717/