Lambda Services for Cloud

ECE 4150 Cloud Computing

Spring 2024

Vijay Madisetti

Georgia Tech

Introduction to Serverless PaaS

How Lambda Services work

How you can realize cloud applications using Lambda Services

Evolution to Function-as a-Service (FaaS)

Model 1: Users execute code on their local servers (prior to cloud)

Model 2: Users can execute code on remote servers by logging in (ssh or rlogin) and running code and databases on remove server.

Model 3: Users provide code to Amazon and AWS runs it for one when your customers or users' applications need it to run and users access the results over the network.

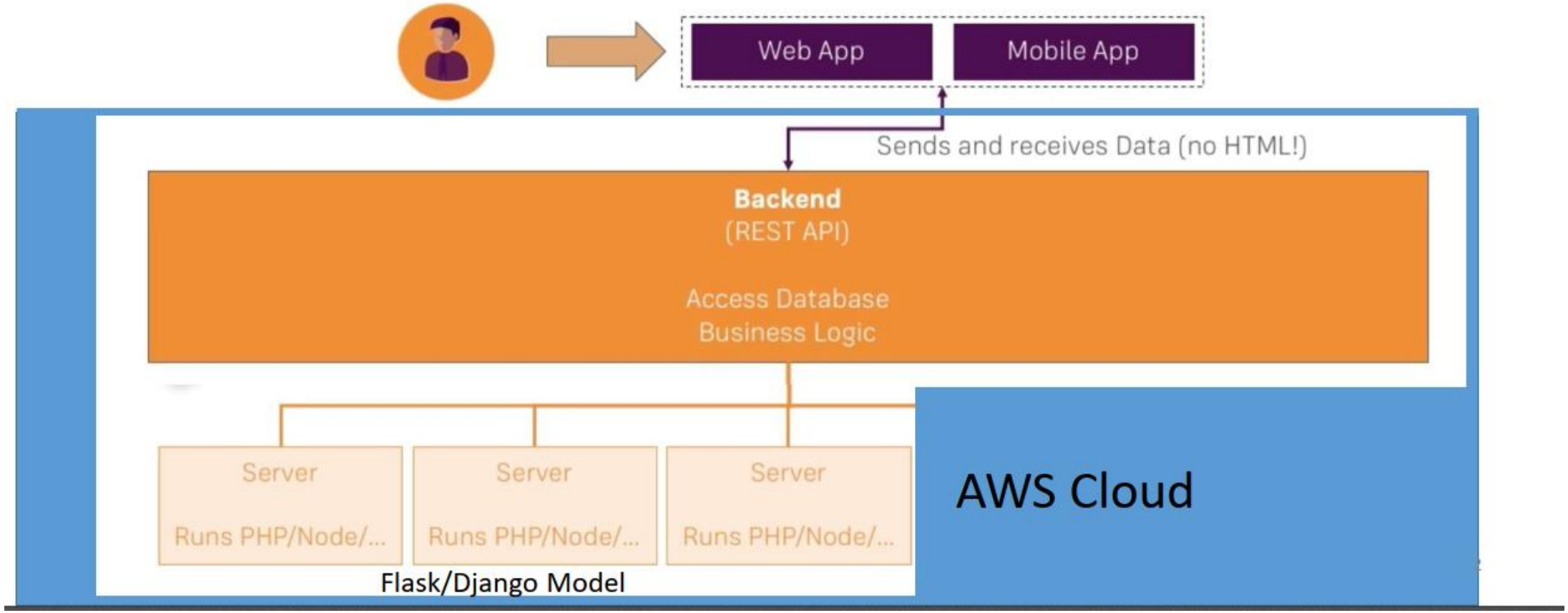
Lambda Services

HOW DOES AWS EXECUTE YOUR CODE FOR YOU?

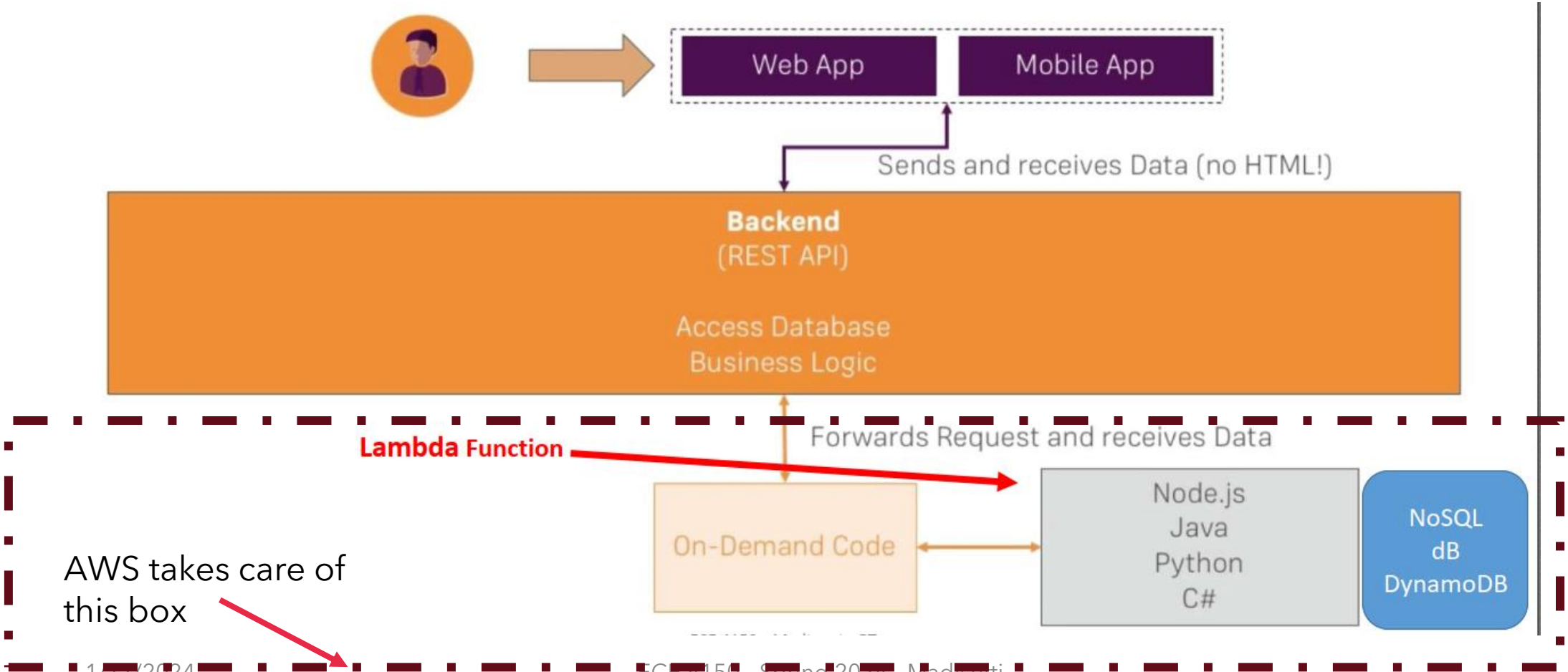
WHY IS IT CALLED “SERVERLESS” ?

DOES MICROSOFT AZURE OFFER THIS FEATURE?

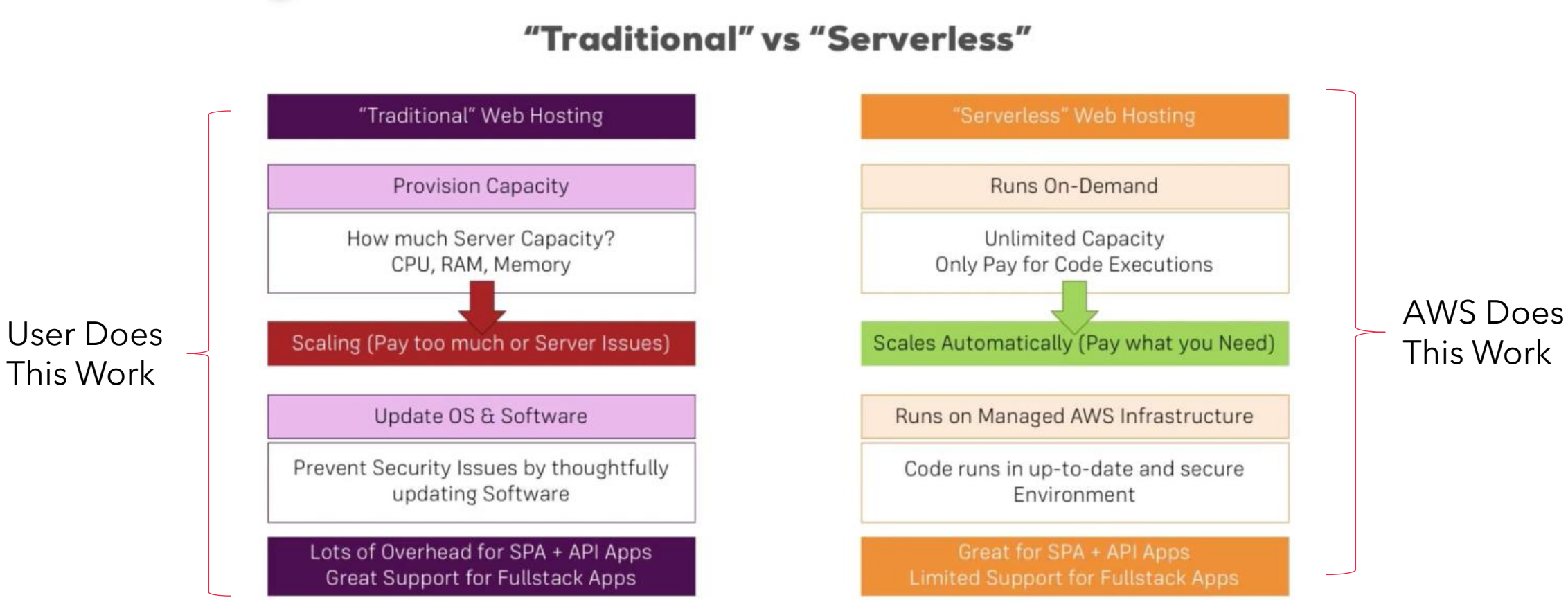
Traditional Web-based Services Model



Serverless Model



AWS does most of the work!



A Few Happy Customers of Lambda Services

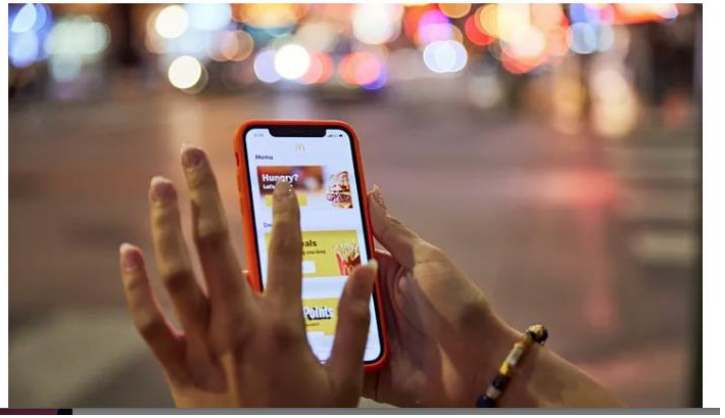
Customers

airbnb
AdRoll
ALTS
asurion
Atlassian
Benchling
BUSTLE
CITRIX
Coca-Cola
COMCAST
Discovery CHANNEL
Expedia
Experian
FireEye
instacart
LexisNexis
Localytics
McGraw Hill
mlbam
New York Public Library
Nextdoor
SCHOLASTIC
The Seattle Times
The Honest Co.
THOMSON REUTERS
vevo
VidRoll
ZAPPROVED
Zillow
23andMe

Enhancing Loyalty rewards: How McDonald's leverages AWS Lambda for microservices

Global Technology · Follow
Published in McDonald's Technical Blog · 4 min read · Nov 7, 2023

43 2



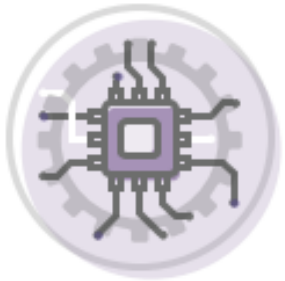
Common Use Cases for AWS Lambda Services

Web Applications

- Static websites
- Complex web apps
- Packages for Flask and Express

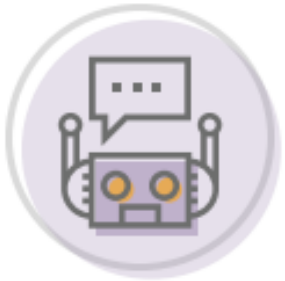
Backends

- Apps & services
- Mobile
- IoT



Data Processing

- Real time
- MapReduce
- Batch

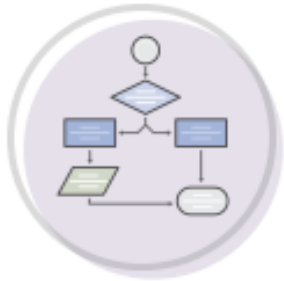


Chatbots

- Powering chatbot logic

Amazon Alexa

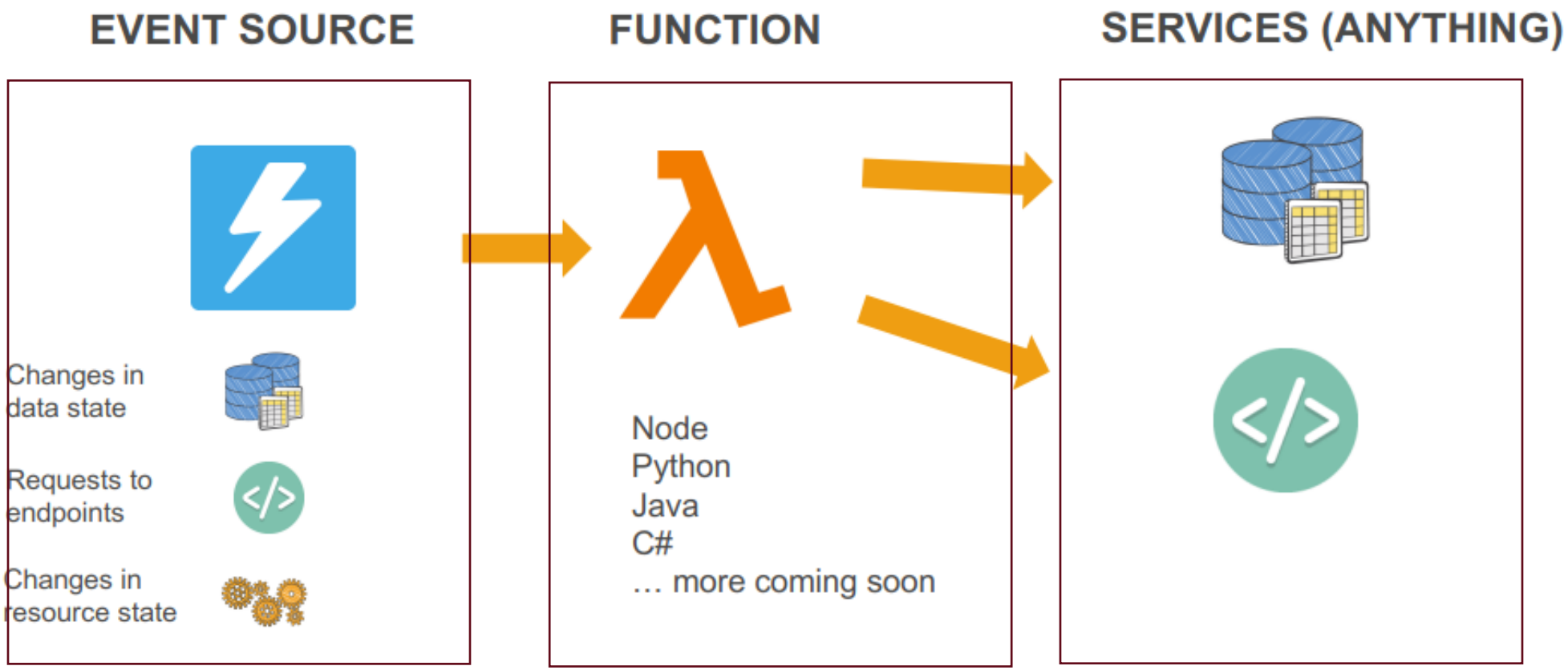
- Powering voice-enabled apps
- Alexa Skills Kit



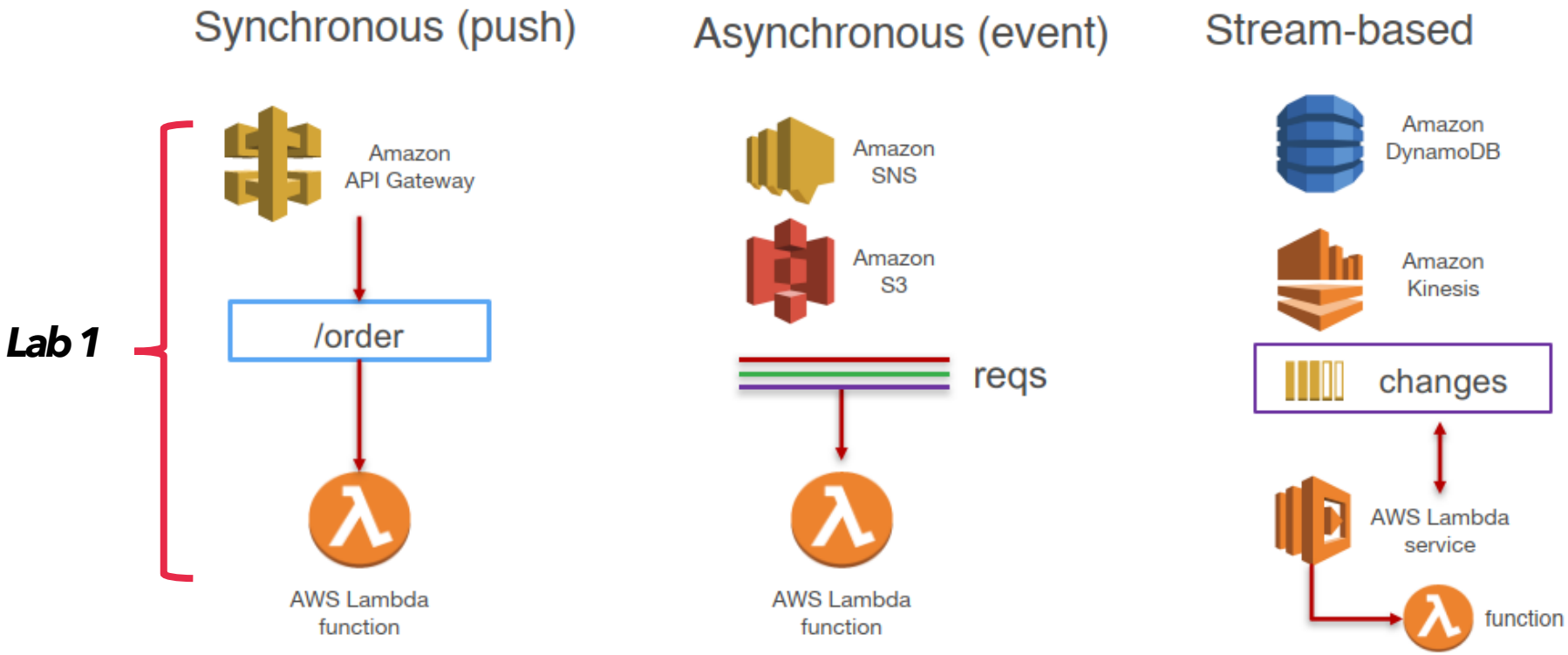
Autonomous IT

- Policy engines
- Extending AWS services
- Infrastructure management

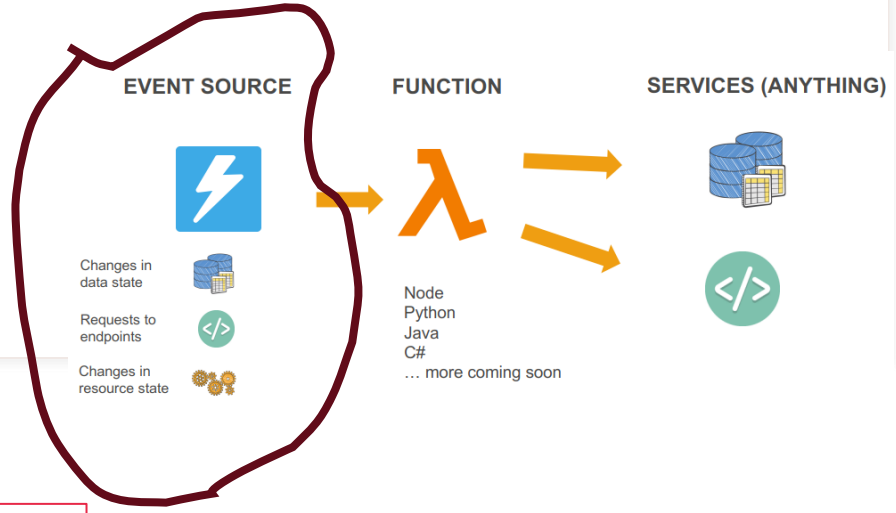
How AWS Lambda Works



Lambda Execution Models



Event Sources Partial List



Anatomy of a Lambda function

Handler() function

Function to be executed upon invocation

Event object

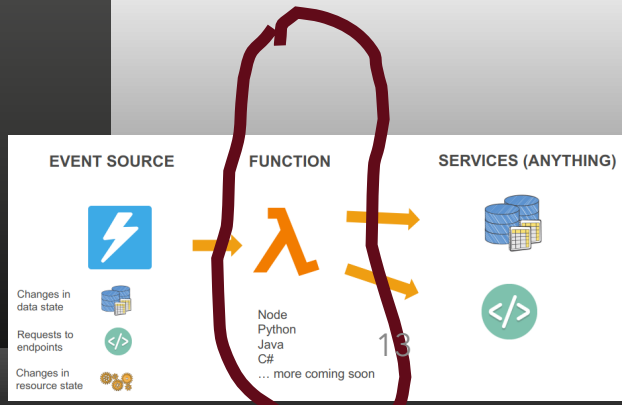
Data sent during Lambda Function Invocation

Context object

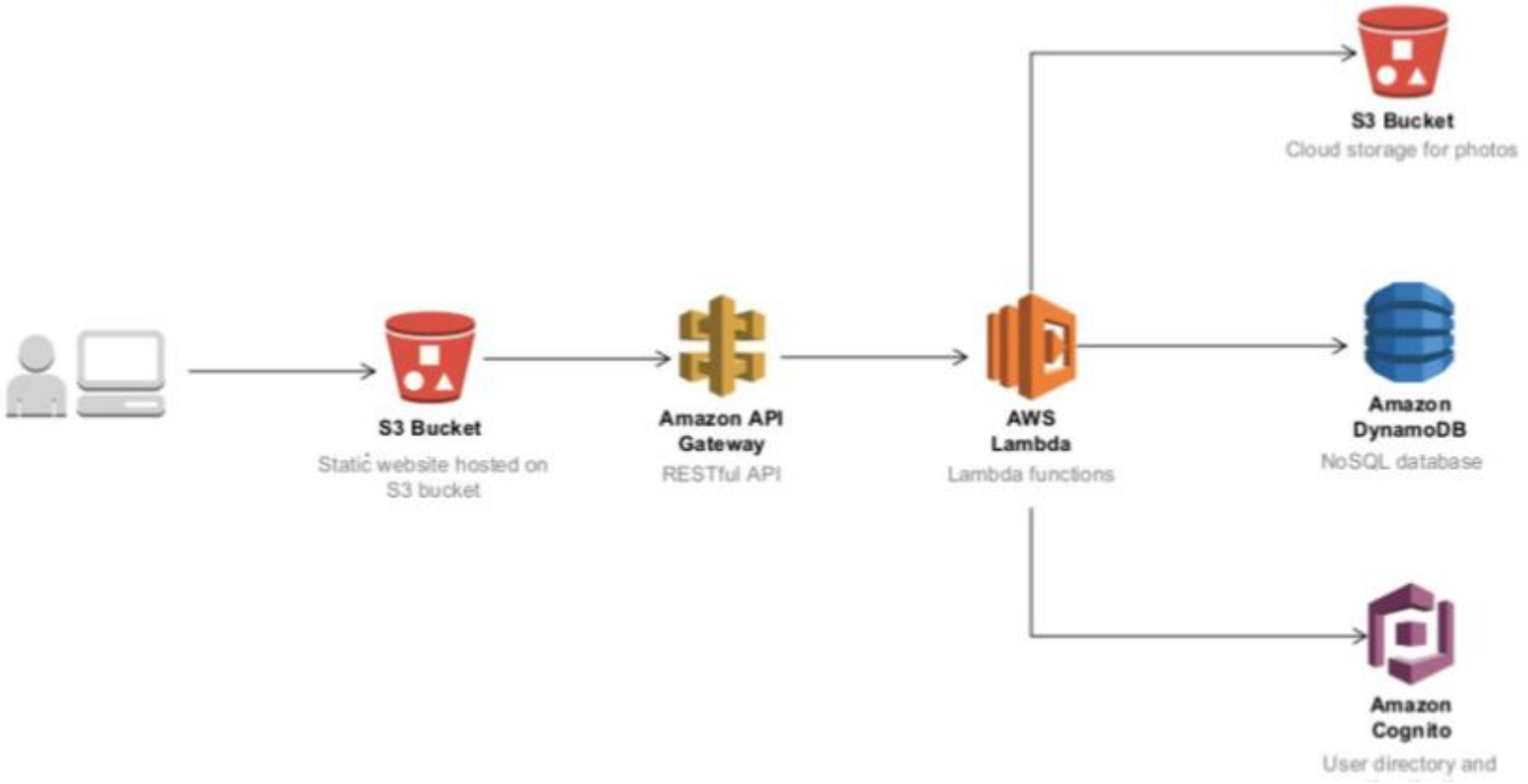
Methods available to interact with runtime information (request ID, log group, etc.)

Lambda Function

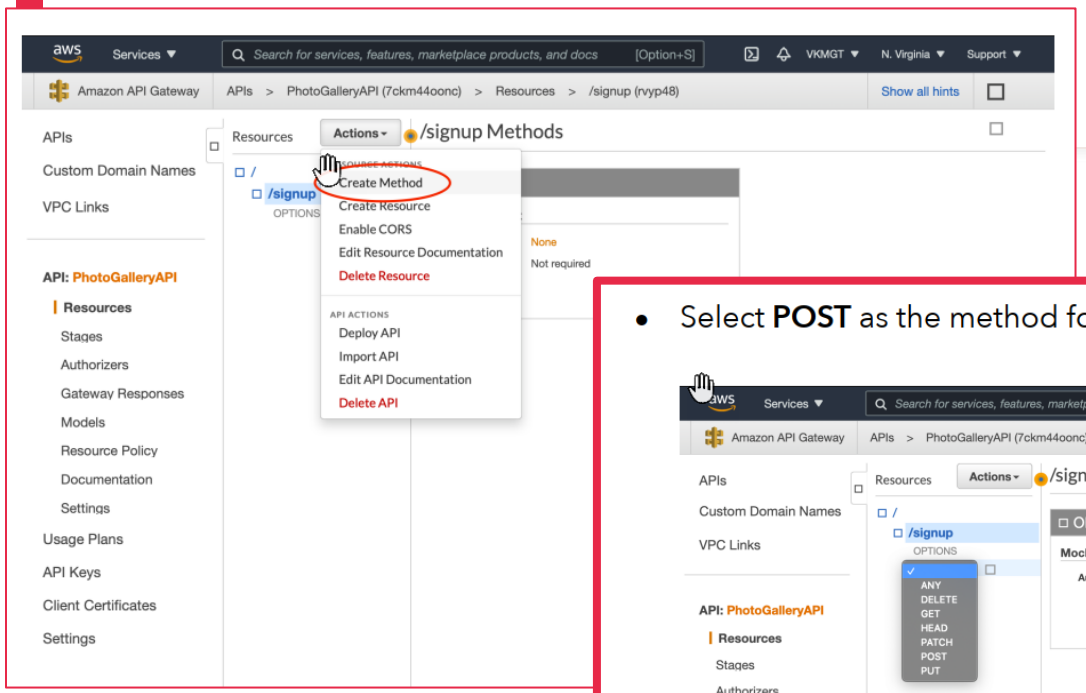
```
s3 = boto3.resource('s3')
app = App()

def lambda_handler(event, context):
    # do something
    ...
```



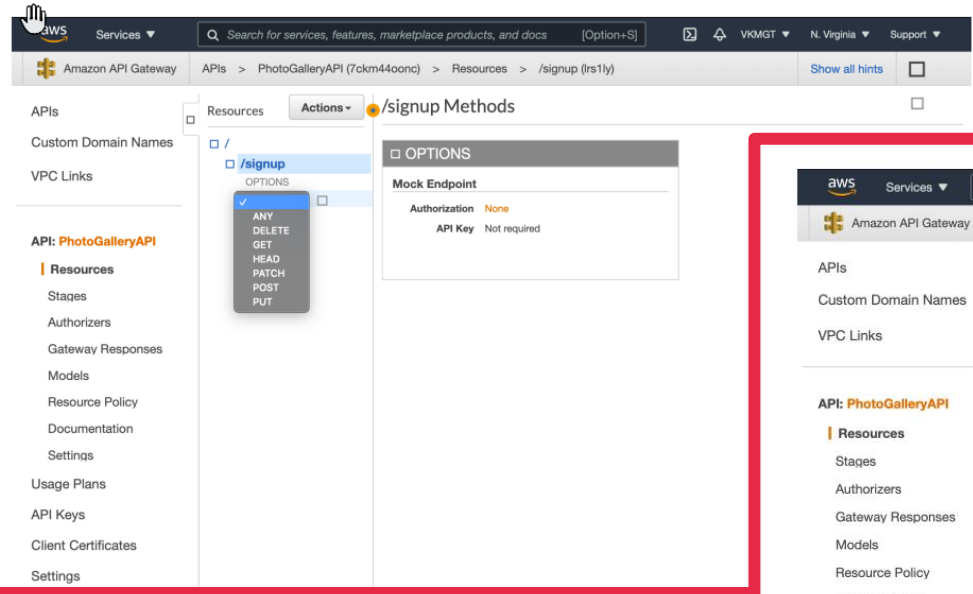
Lab 1 - Photo Gallery

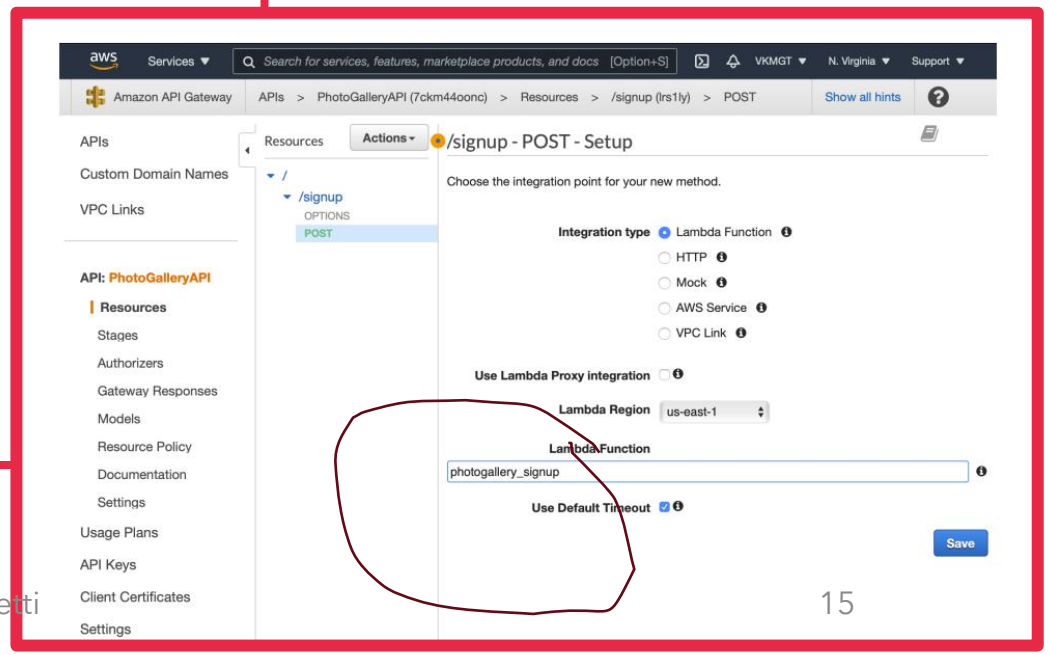


How to Bind A Function to a Method

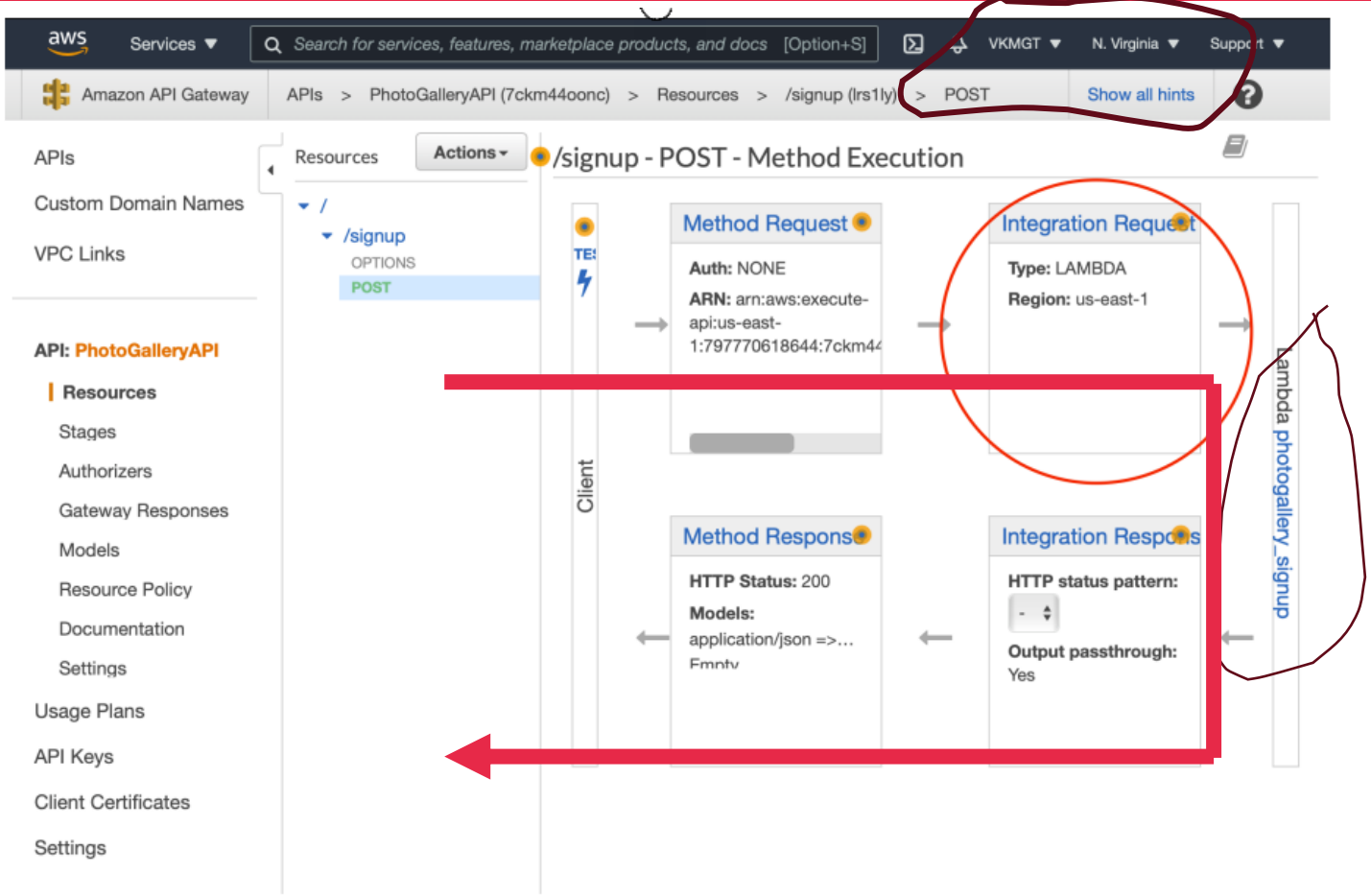


- Select **POST** as the method for this resource

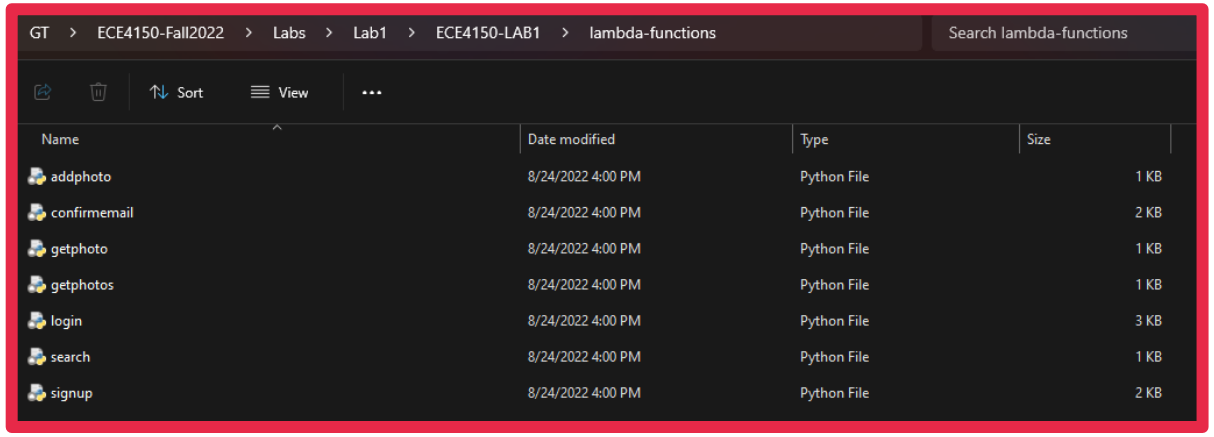




Round Trip


aws
Services
Search for services, features, marketplace products, and docs [Option+S]
VKMGT
N. Virginia
Amazon API Gateway
APIs > PhotoGalleryAPI (7ckm44oonc) > Resources > /signup (lrs1ly) > POST
Show all hints
APIs
Custom Domain Names
VPC Links
API: PhotoGalleryAPI
Resources
Stages
Authorizers
Gateway Responses
Models
Resource Policy
Documentation
Settings
Usage Plans
API Keys
Client Certificates
Settings
Resources
Actions
/signup - POST - Method Execution
/
/signup
OPTIONS
POST
Client
Method Request
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:797770618644:7ckm44
Integration Request
Type: LAMBDA
Region: us-east-1
Lambda photogallery_signup
Method Response
HTTP Status: 200
Models:
application/json =>...
Empty
Integration Response
HTTP status pattern:
Output passthrough:
Yes

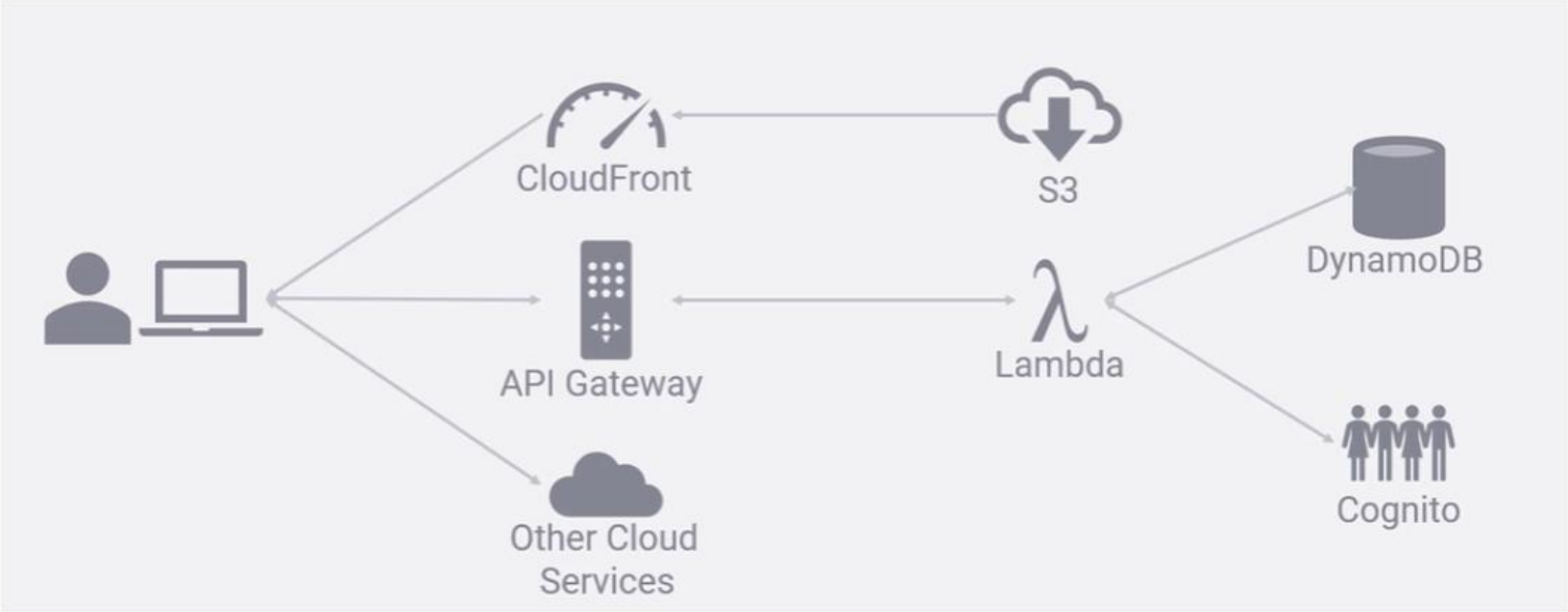
Lambda Function – *Signup* Microservice



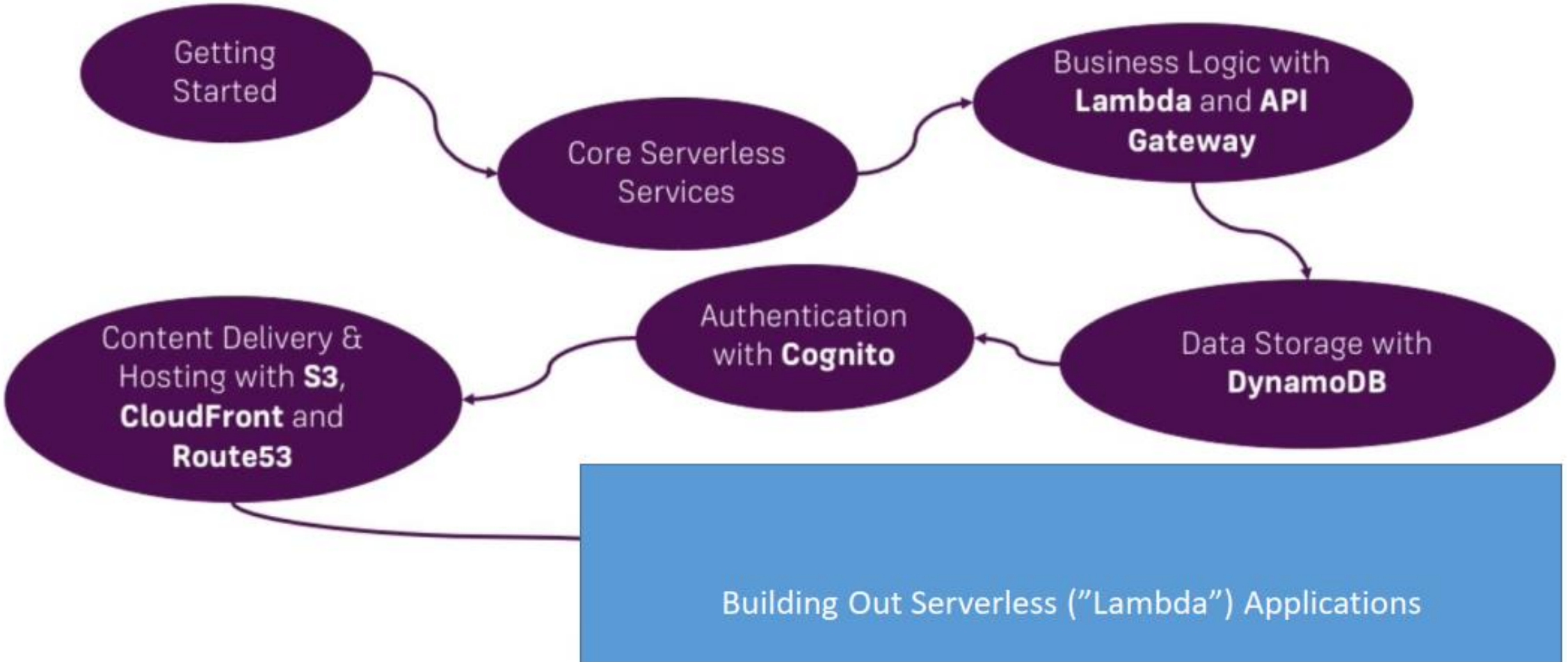

```python
import json
import boto3
from botocore.exceptions import ClientError

REGION="us-east-1"
USER_POOL_ID="us-east-1_uFiTfeMfP"
CLIENT_ID="347k41i5qo5qrmoclko9n19daa"

cognitoclient = boto3.client('cognito-idp', region_name=REGION)

def lambda_handler(event, context):
    username=event['body-json']['username']
    password=event['body-json']['password']
    name=event['body-json']['name']
    email=event['body-json']['email']
    result=False
    message=""
    response={}
    returndata={}
    userdata={}

    try:
        response = cognitoclient.sign_up(
            ClientId=CLIENT_ID,
            Username=username,
            Password=password,
            UserAttributes=[
                {
                    'Name': 'name',
                    'Value': name
                },
                {
                    'Name': 'email',
                    'Value': email
                }
            ]
        )
        result=True
        message="Signup successful"
```

Pattern for Microservices

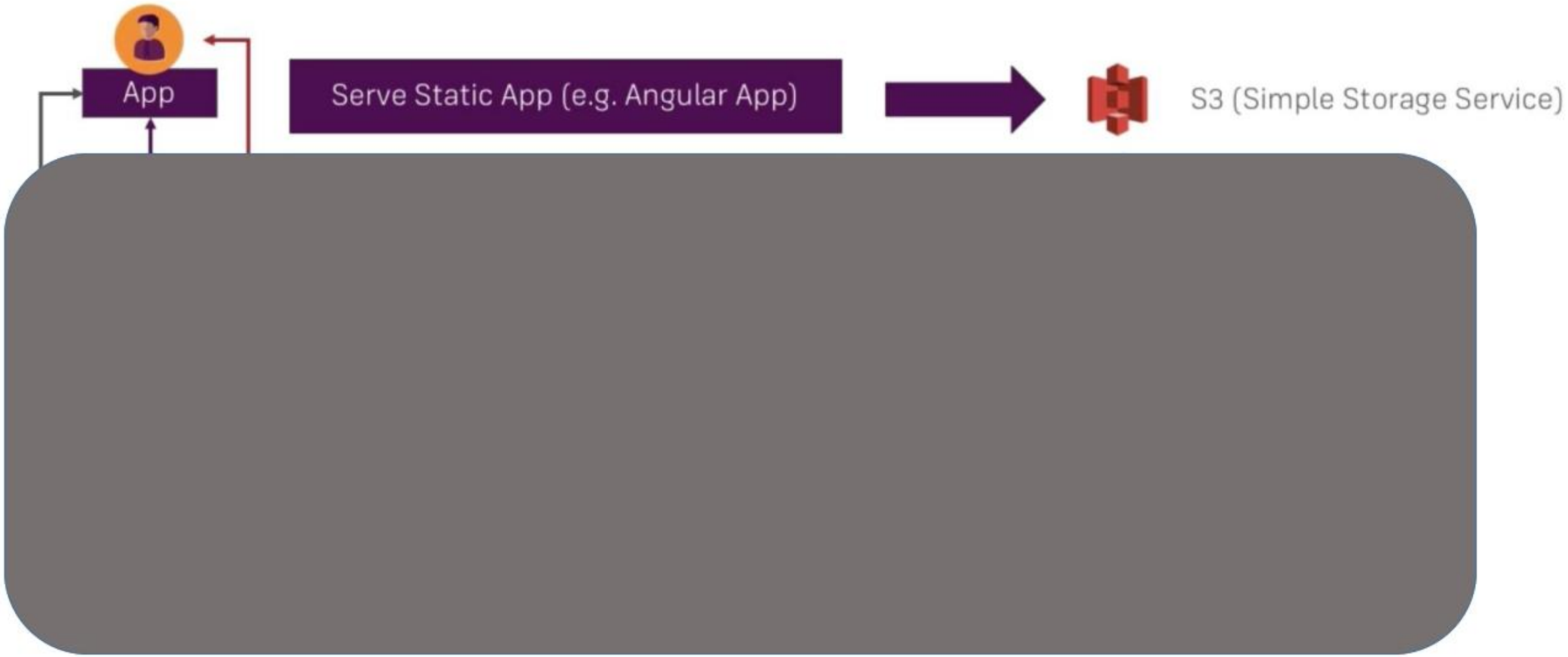
Architecture of a Microservices Application Using Lambda Services



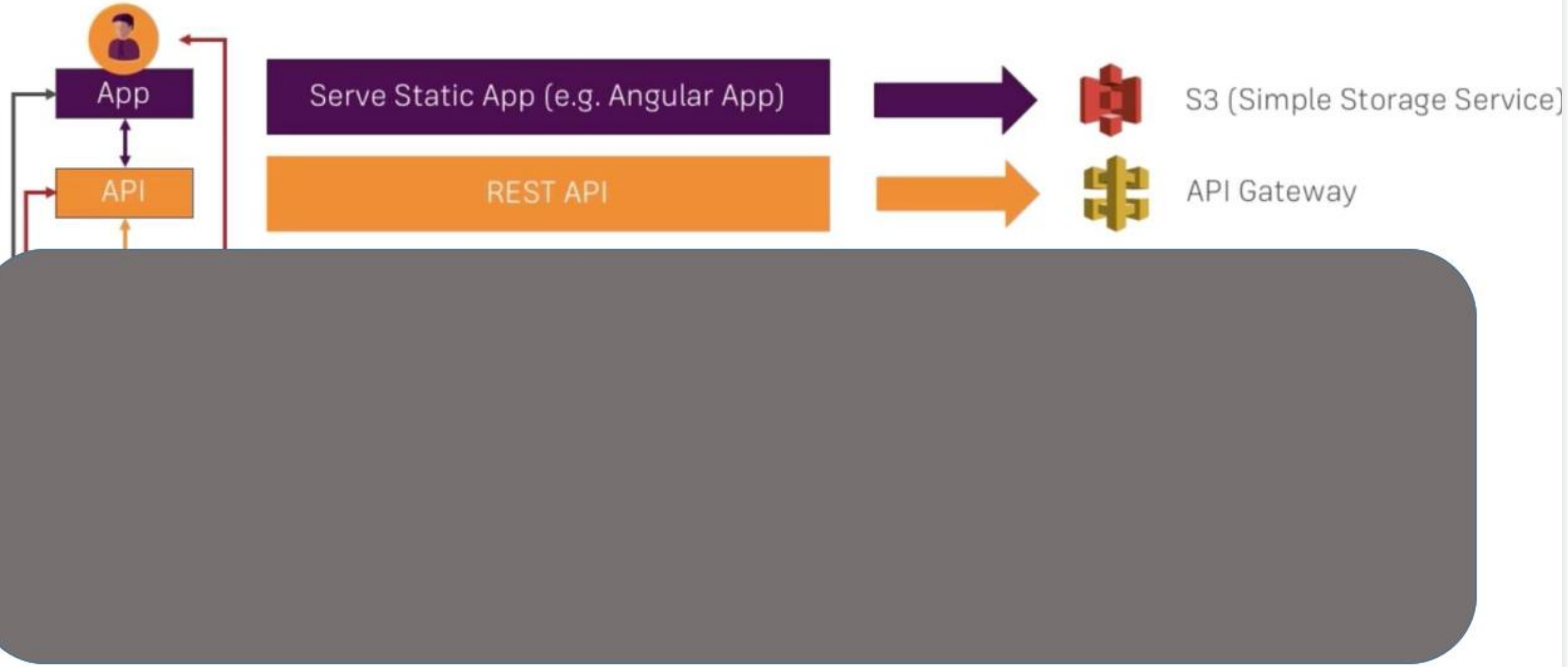
General Development Flow for AWS Lamba



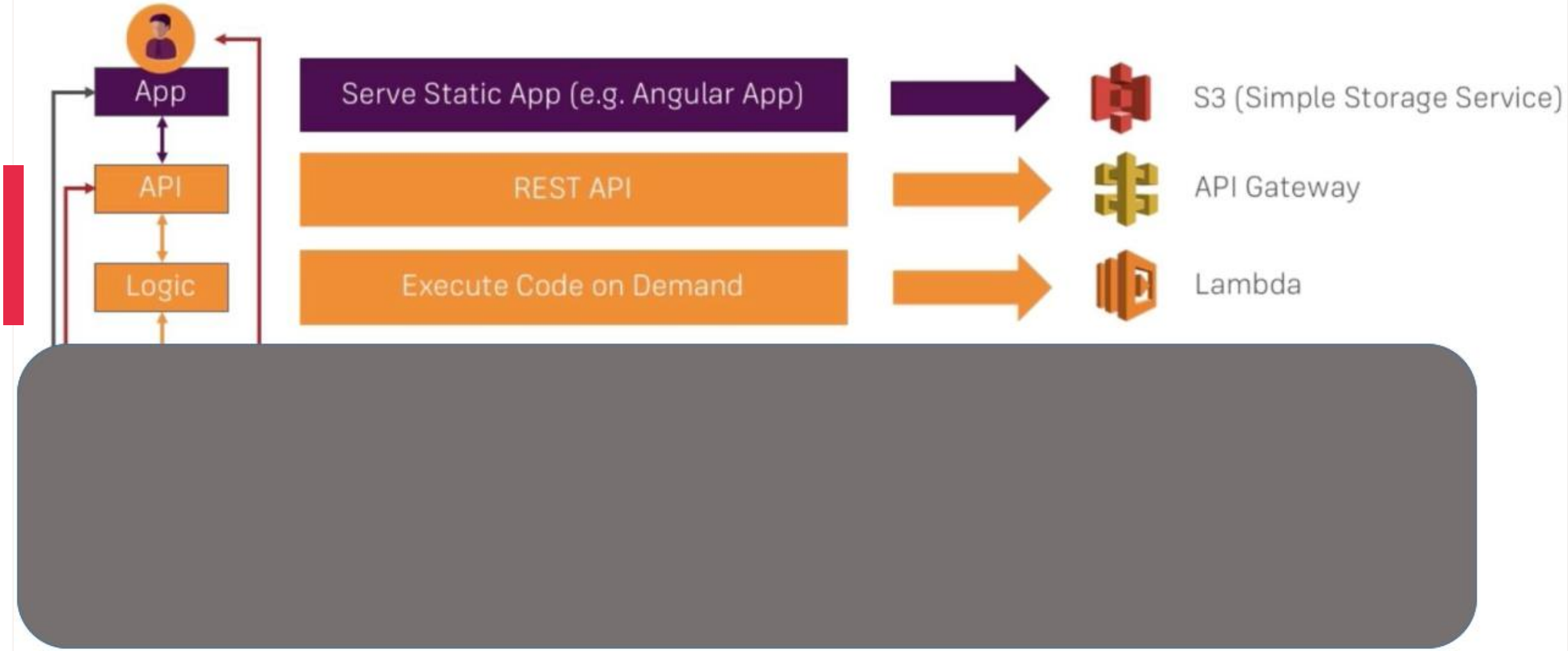
Which Services Do We Need?



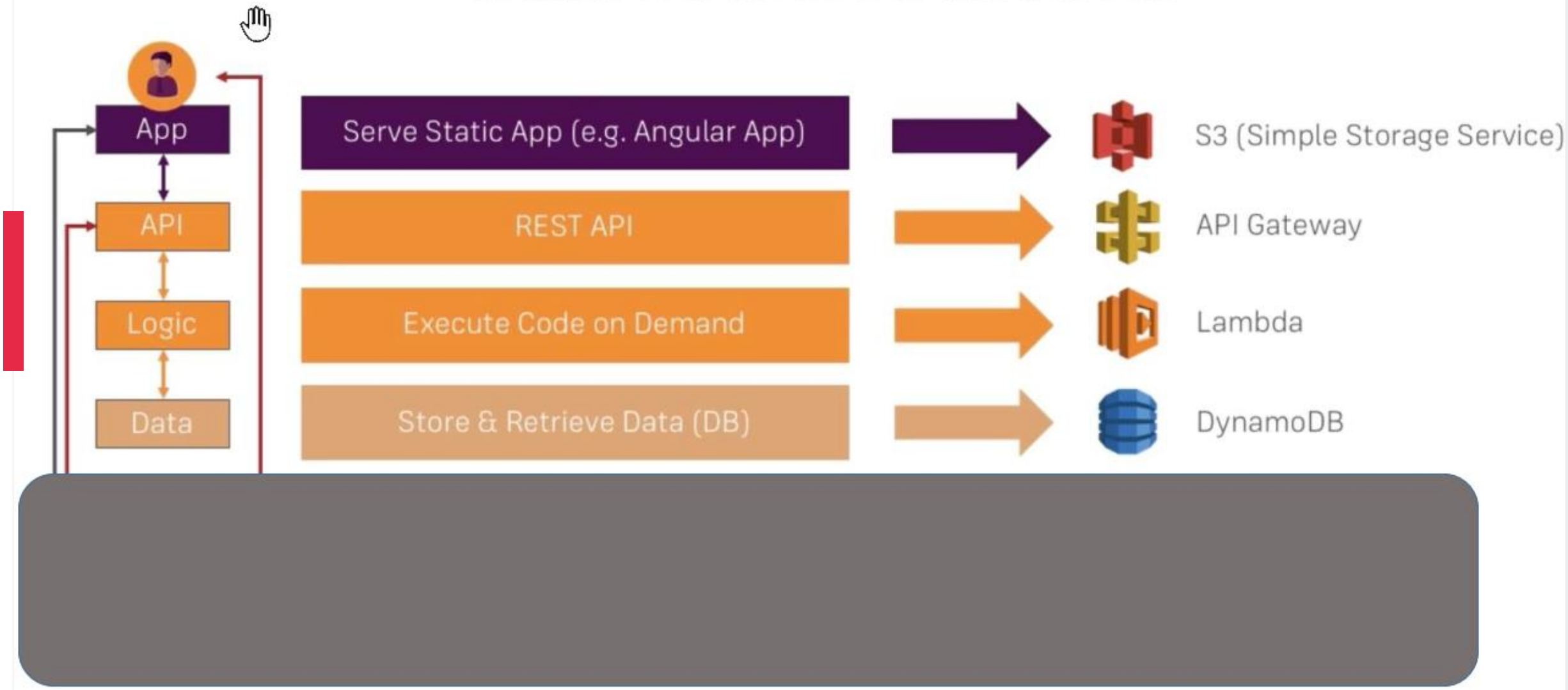
Which Services Do We Need?



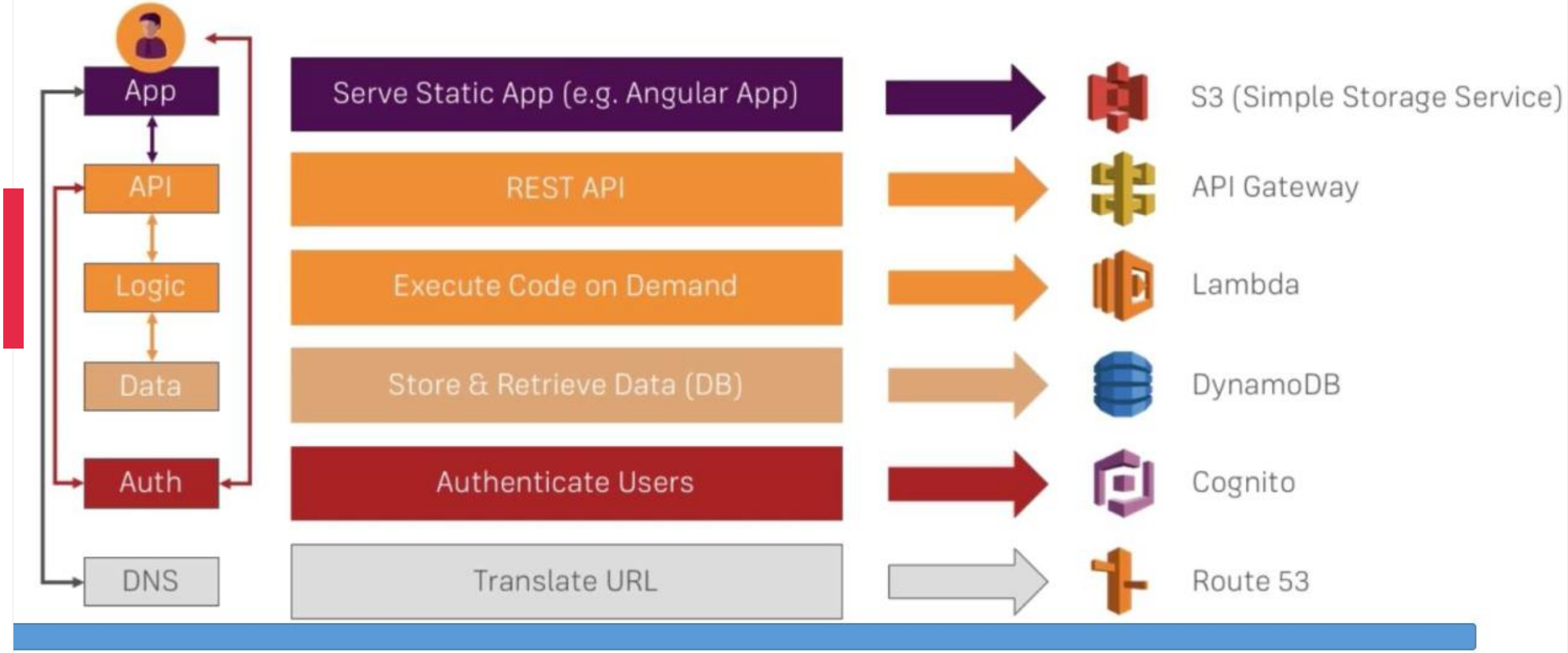
Which Services Do We Need?



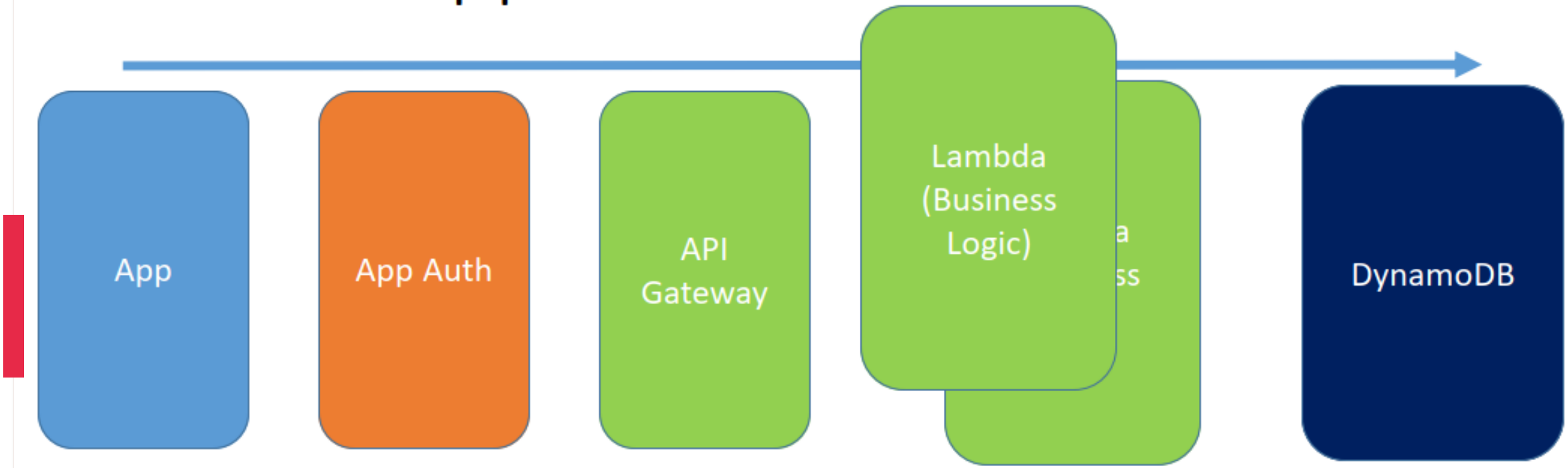
Which Services Do We Need?

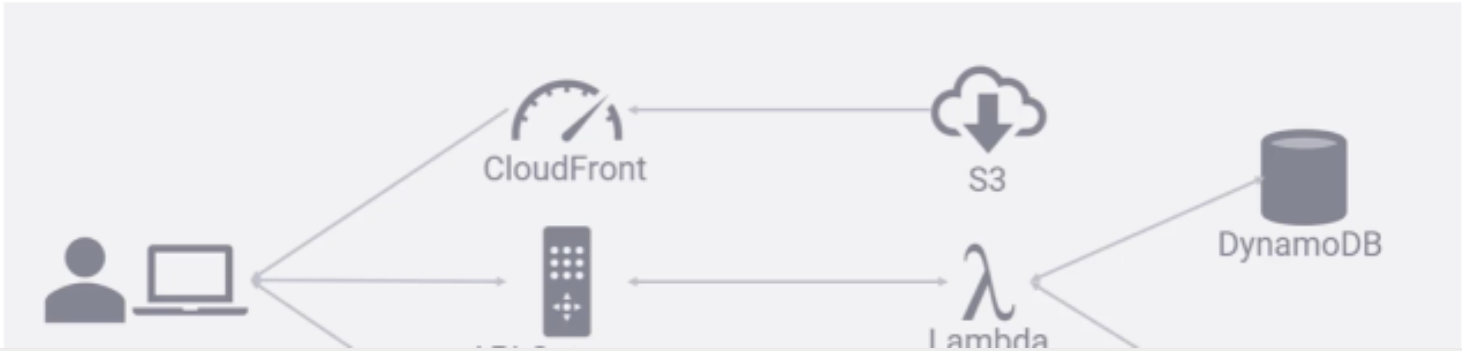


Which Services Do We Need?

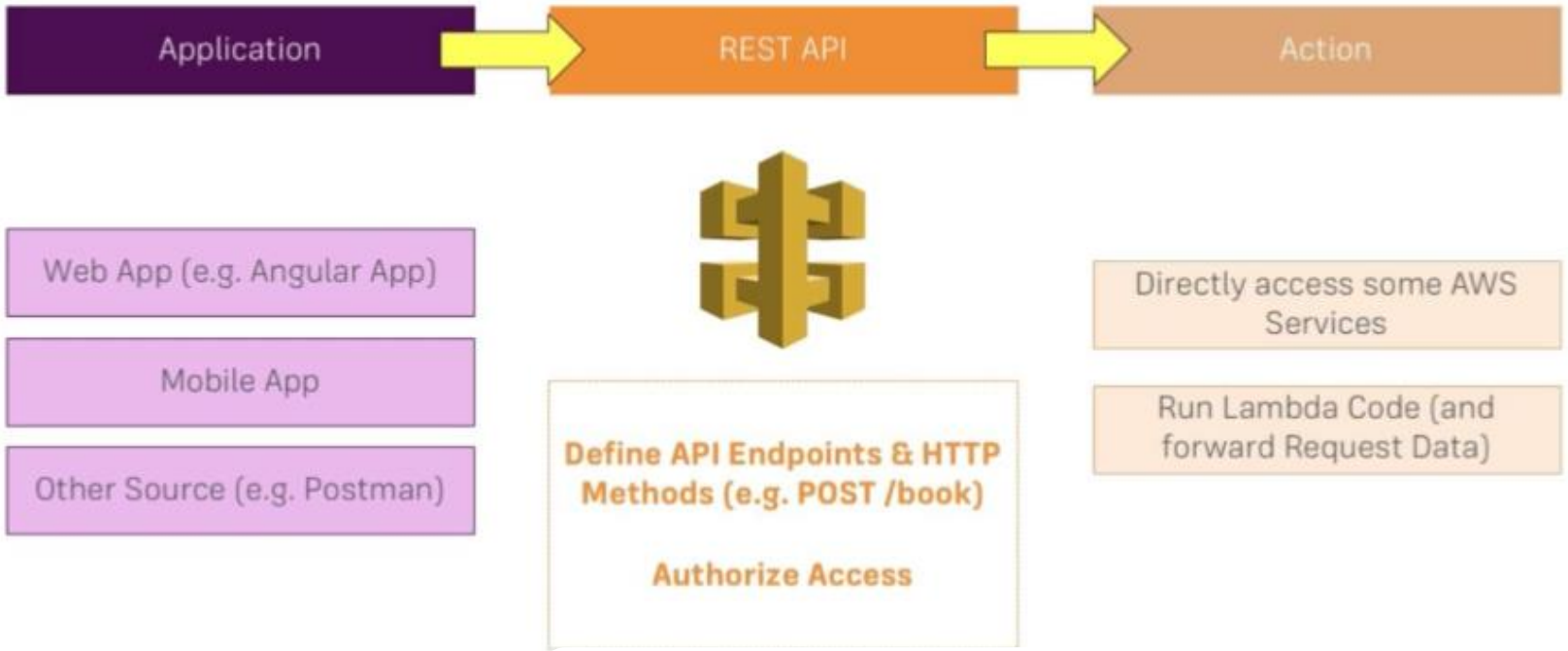


Serverless App Flow

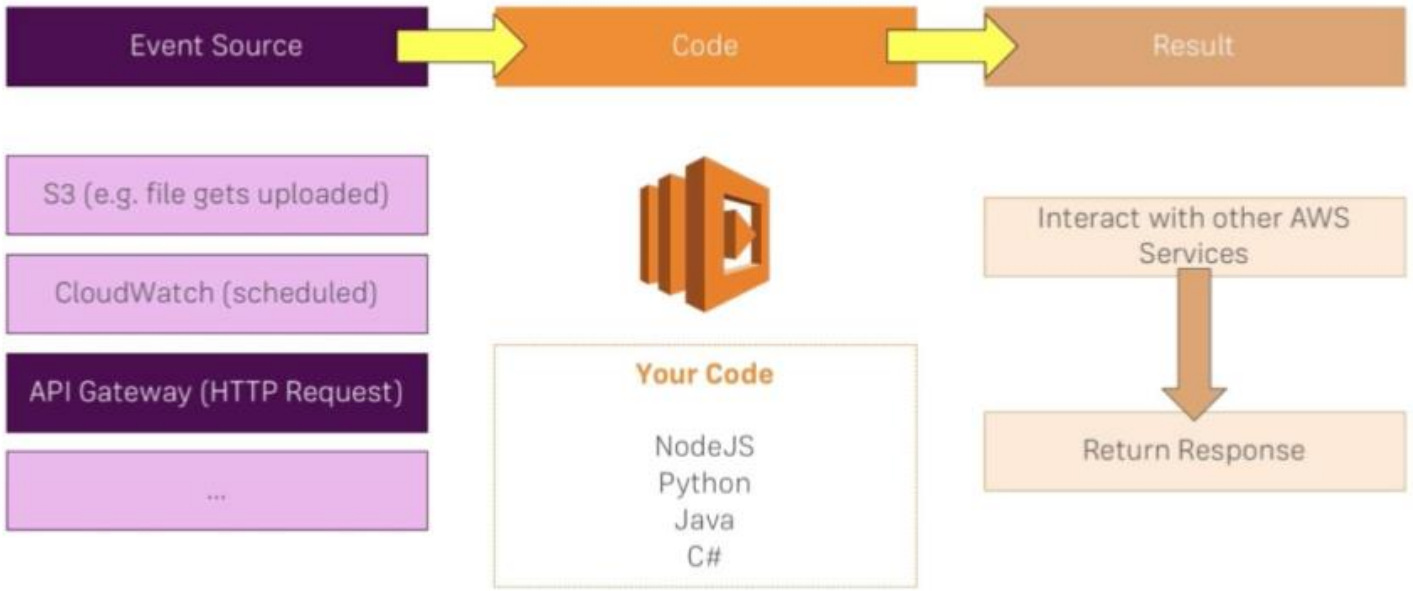




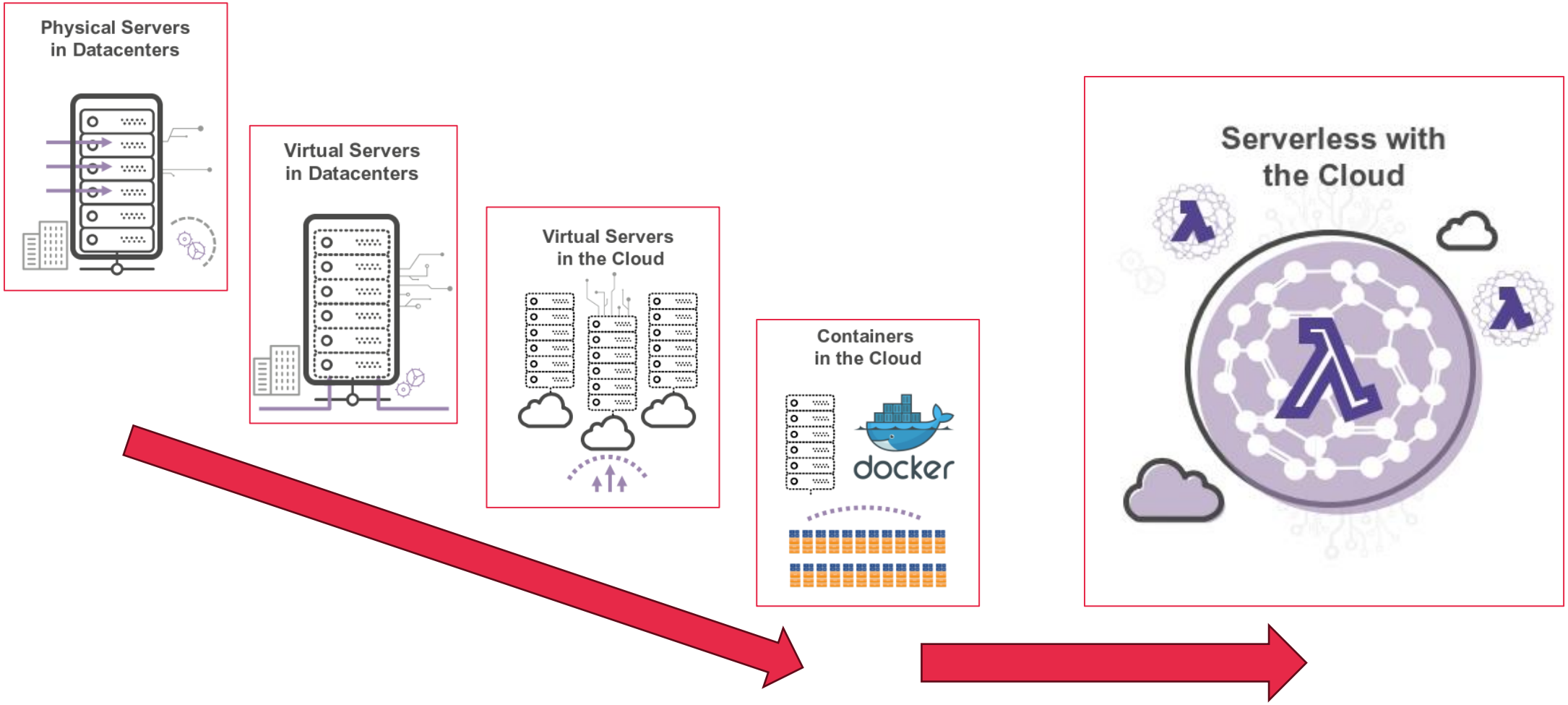
How it works


Application
REST API
Action
Web App (e.g. Angular App)
Mobile App
Other Source (e.g. Postman)
Define API Endpoints & HTTP Methods (e.g. POST /book)
Authorize Access
Directly access some AWS Services
Run Lambda Code (and forward Request Data)

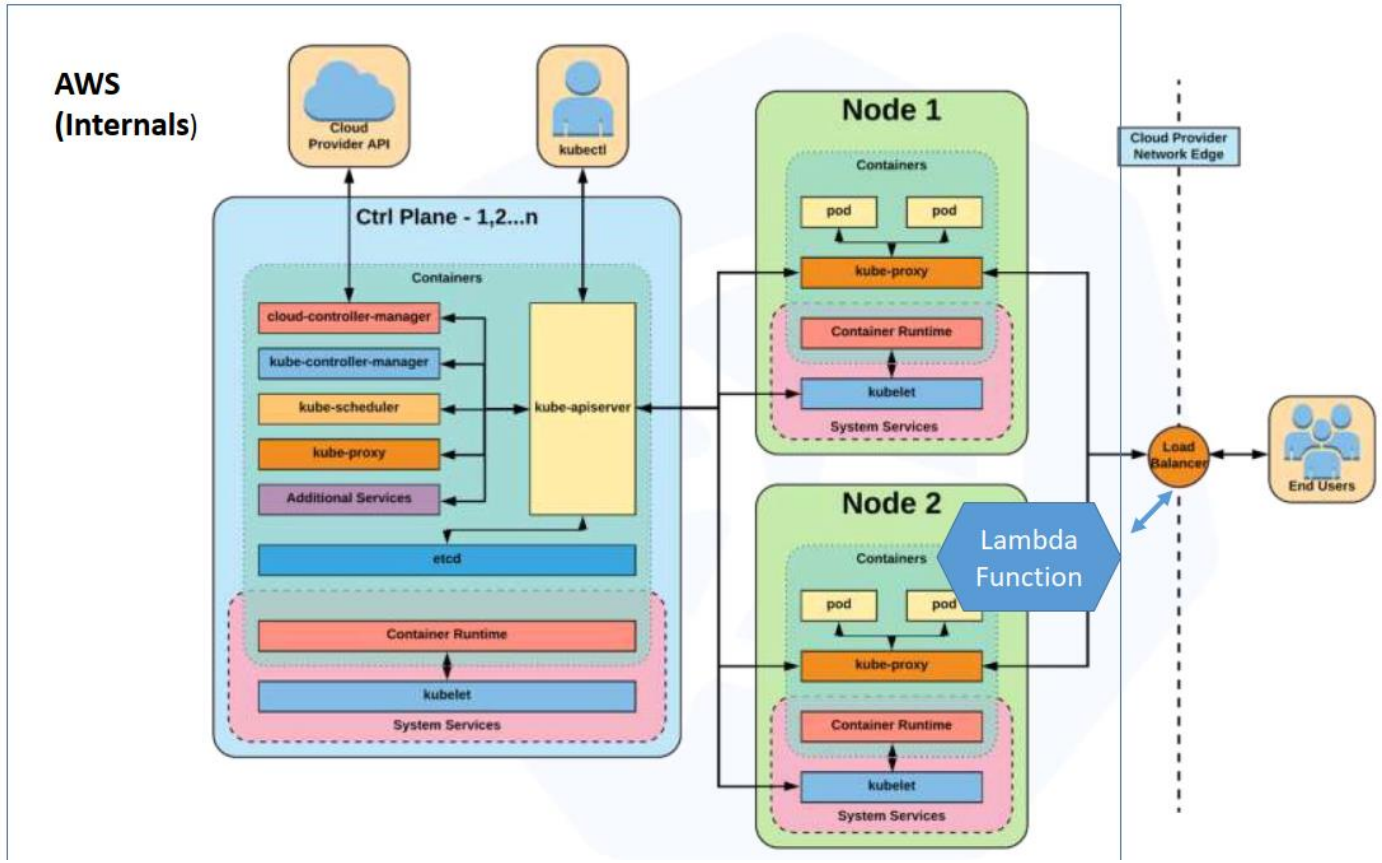
How it works

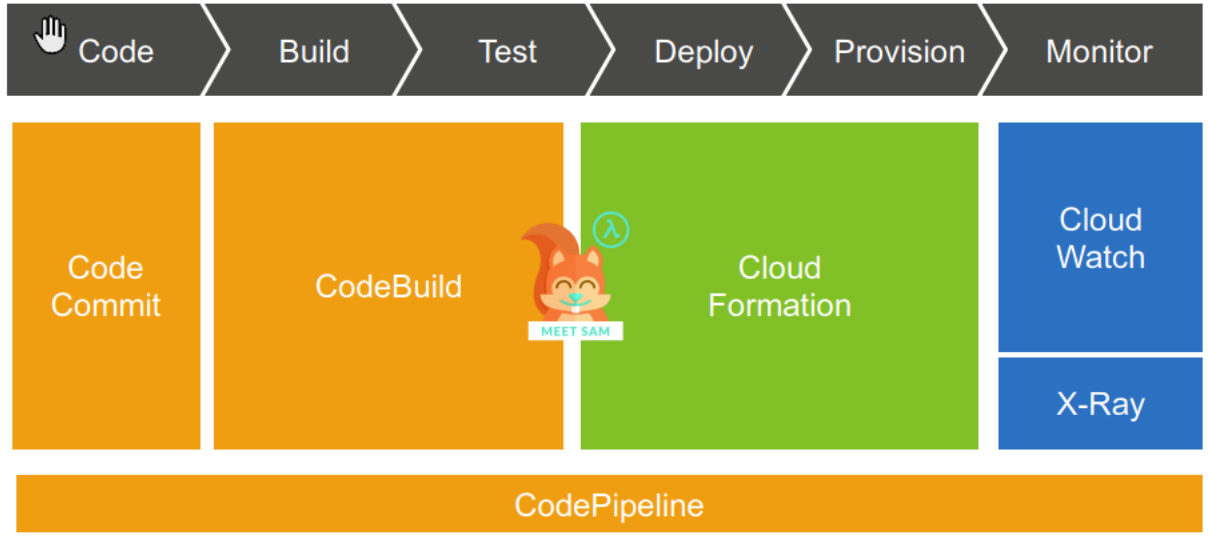

Event Source
Code
Result
S3 (e.g. file gets uploaded)
CloudWatch (scheduled)
API Gateway (HTTP Request)
...
Your Code
NodeJS
Python
Java
C#
Interact with other AWS Services
Return Response

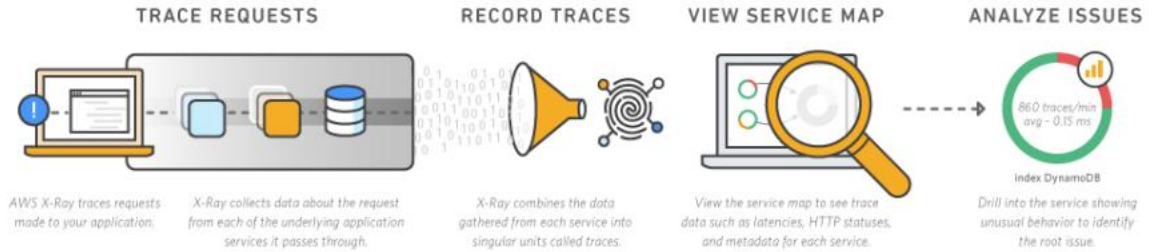
Serverless progression

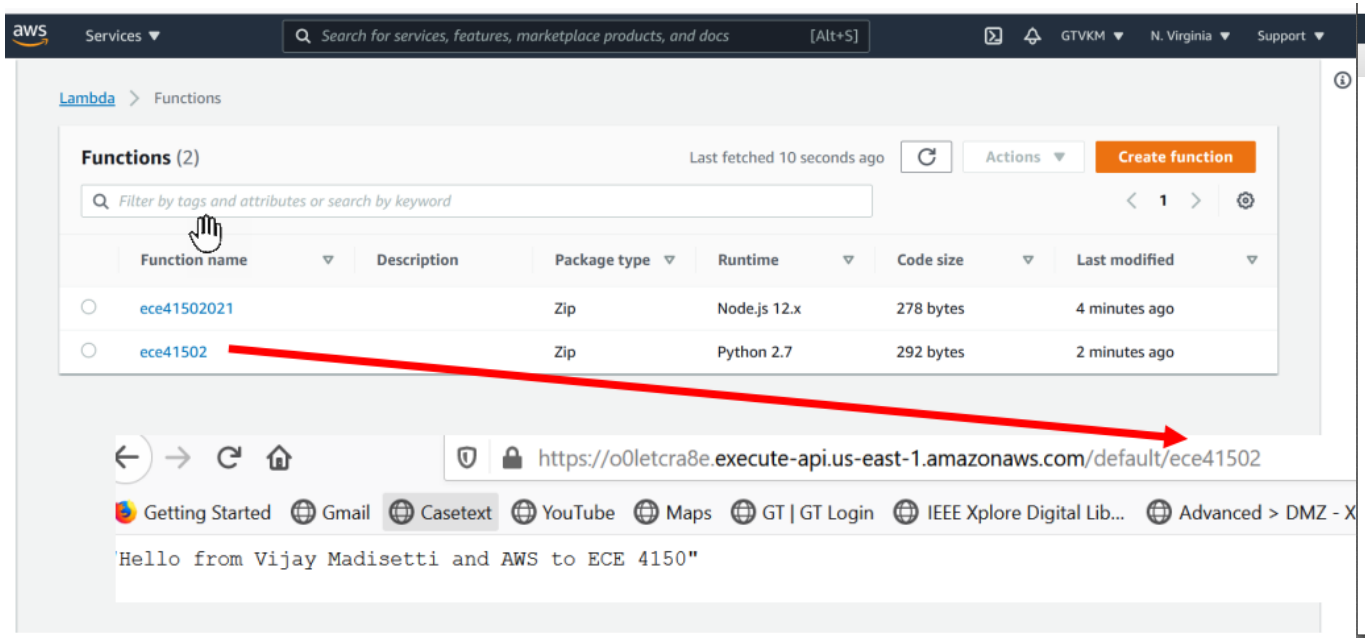


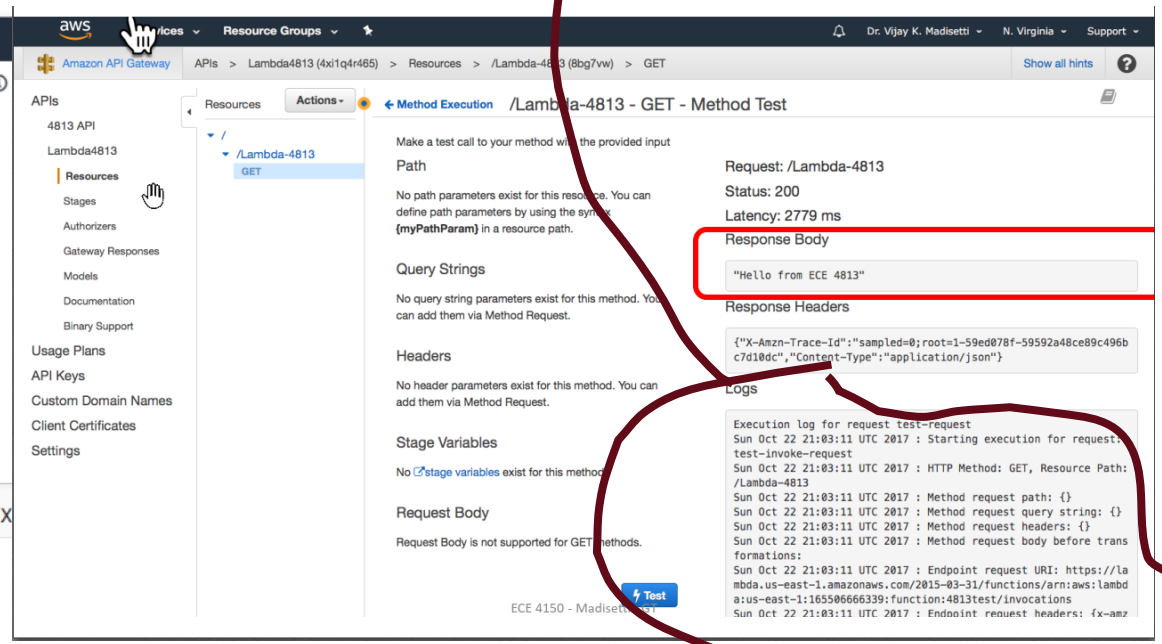
Inside AWS – Kubernetes (to be covered later)




Code
Build
Test
Deploy
Provision
Monitor
Code Commit
CodeBuild
MEET SAM
Cloud Formation
Cloud Watch
X-Ray
CodePipeline

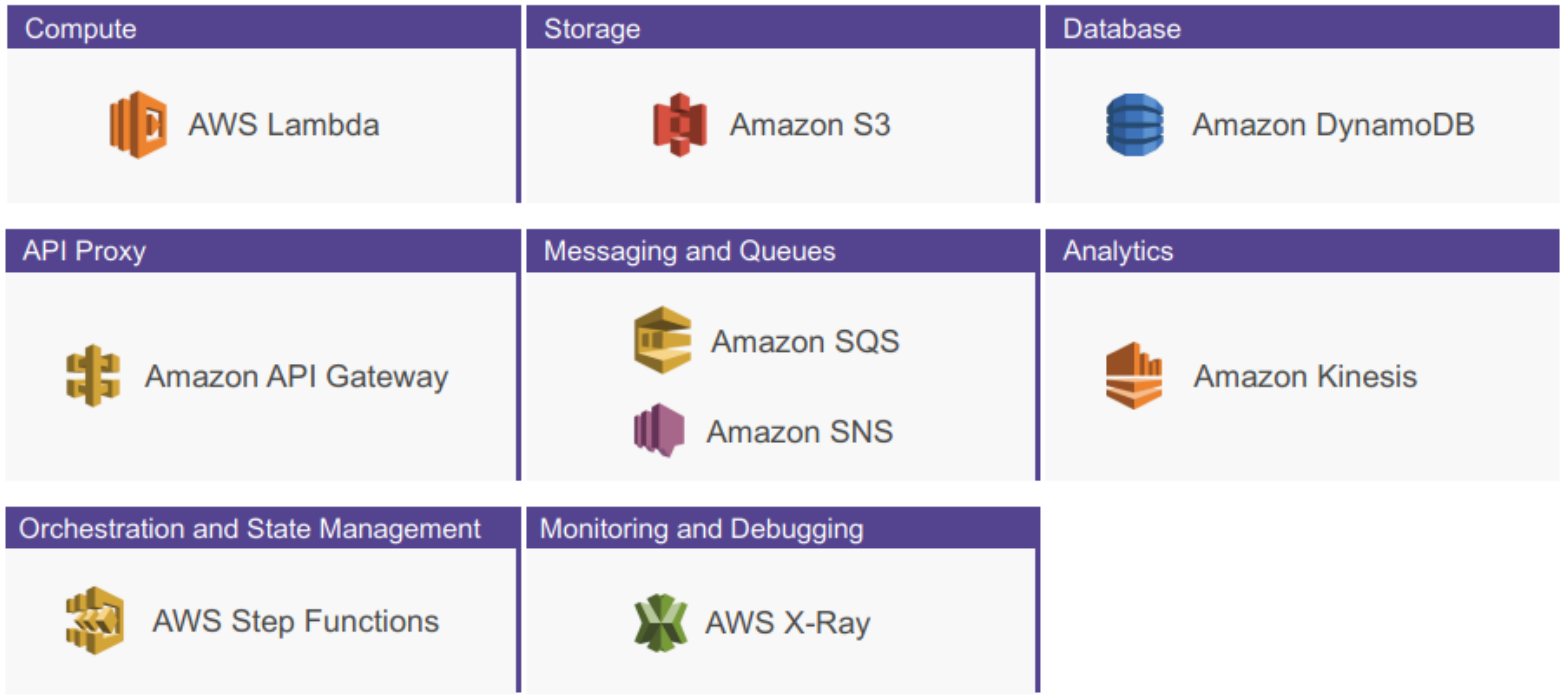

TRACE REQUESTS
RECORD TRACES
VIEW SERVICE MAP
ANALYZE ISSUES
AWS X-Ray traces requests made to your application.
X-Ray collects data about the request from each of the underlying application services it passes through.
X-Ray combines the data gathered from each service into singular units called traces.
View the service map to see trace data such as latencies, HTTP statuses, and metadata for each service.
Drill into the service showing unusual behavior to identify the root issue.
index DynamoDB


Services
Search for services, features, marketplace products, and docs [Alt+S]
GTVKM
N. Virginia
Support
Lambda > Functions
Functions (2)
Last fetched 10 seconds ago
Actions
Create function
Filter by tags and attributes or search by keyword
Function name
Description
Package type
Runtime
Code size
Last modified
ece41502021
Zip
Node.js 12.x
278 bytes
4 minutes ago
ece41502
Zip
Python 2.7
292 bytes
2 minutes ago
https://o0letcra8e.execute-api.us-east-1.amazonaws.com/default/ece41502
Getting Started
Gmail
Casetext
YouTube
Maps
GT | GT Login
IEEE Xplore Digital Lib...
Advanced > DMZ - X
"Hello from Vijay Madisetti and AWS to ECE 4150"

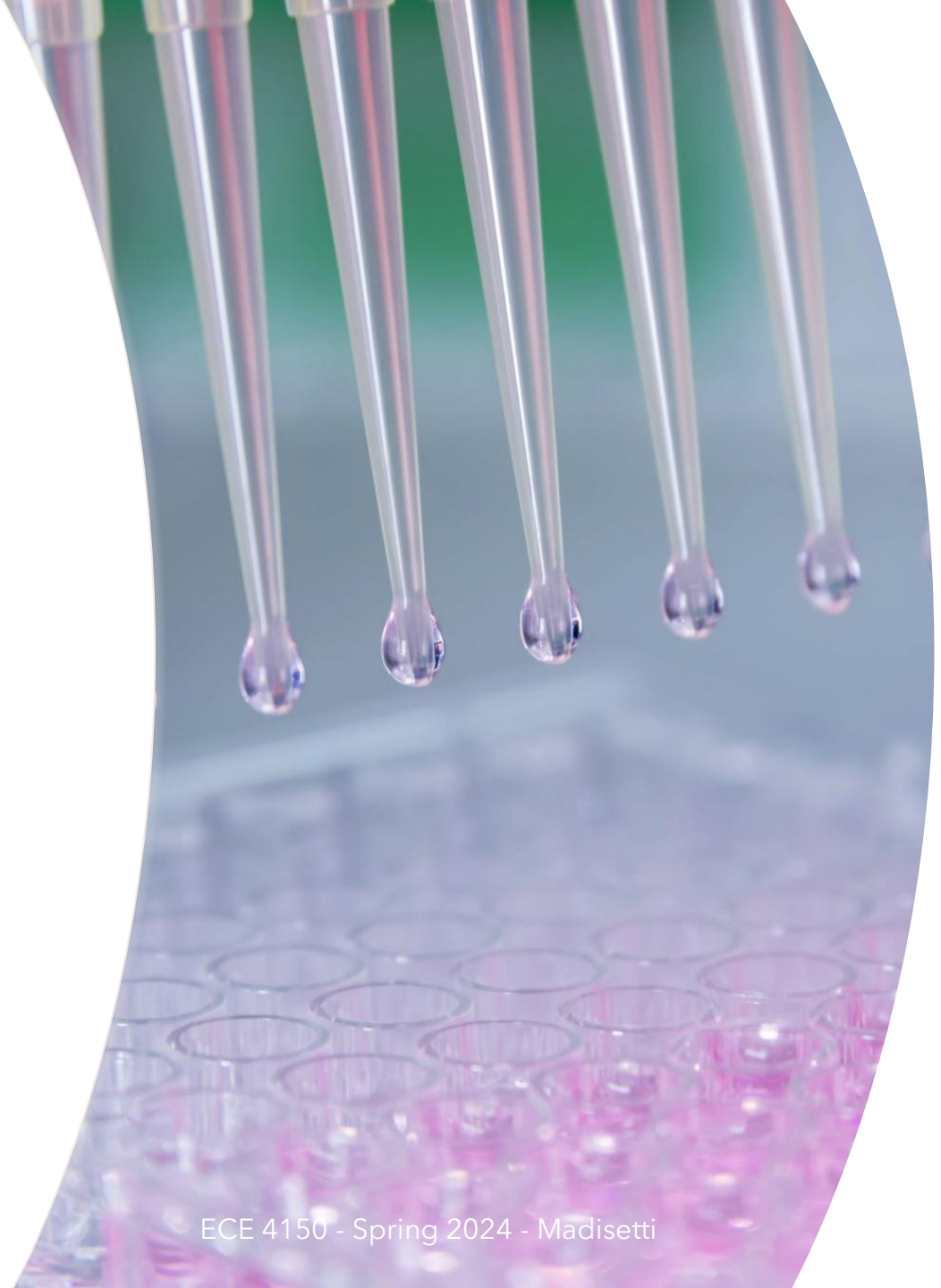

Resource Groups
Dr. Vijay K. Madisetti
N. Virginia
Support
Amazon API Gateway
APIs > Lambda4813 (4xi1q4r465) > Resources > /Lambda-4813 (8bg7vw) > GET
Show all hints
APIs
4813 API
Lambda4813
Resources
Stages
Authorizers
Gateway Responses
Models
Documentation
Binary Support
Usage Plans
API Keys
Custom Domain Names
Client Certificates
Settings
Resources
Actions
/
/Lambda-4813
GET
Method Execution
/Lambda-4813 - GET - Method Test
Make a test call to your method with the provided input
Path
No path parameters exist for this resource. You can define path parameters by using the syntax {myPathParam} in a resource path.
Query Strings
No query string parameters exist for this method. You can add them via Method Request.
Headers
No header parameters exist for this method. You can add them via Method Request.
Stage Variables
No stage variables exist for this method.
Request Body
Request Body is not supported for GET methods.
Test
Request: /Lambda-4813
Status: 200
Latency: 2779 ms
Response Body
"Hello from ECE 4813"
Response Headers
{"X-Amzn-Trace-Id":"sampled=0;root=1-59ed078f-59592a48ce89c496bc7d10dc","Content-Type":"application/json"}
Logs
Execution log for request test-request
Sun Oct 22 21:03:11 UTC 2017 : Starting execution for request: test-invoke-request
Sun Oct 22 21:03:11 UTC 2017 : HTTP Method: GET, Resource Path: /Lambda-4813
Sun Oct 22 21:03:11 UTC 2017 : Method request path: {}
Sun Oct 22 21:03:11 UTC 2017 : Method request query string: {}
Sun Oct 22 21:03:11 UTC 2017 : Method request headers: {}
Sun Oct 22 21:03:11 UTC 2017 : Method request body before transformations:
Sun Oct 22 21:03:11 UTC 2017 : Endpoint request URI: https://lambda.us-east-1.amazonaws.com/2015-03-31/functions/arn:aws:lambda:us-east-1:165506666339:function:4813test/invocations

Building Blocks

Building blocks for serverless applications



Summary



- **AWS Lambda services are powerful and convenient to harness the cloud**
- **Lab 1 will develop these concepts into "practice".**