

AWS Developer: Lambda Deep Dive

GETTING STARTED WITH AWS LAMBDA



Dror Helper

@dhelper www.helpercode.com





Who is this course for?



Cloud Developers and Architects

- New and veteran Cloud developers
- Learn to harness the power AWS Lambda
- Create event-driven, serverless architecture
- Create workflows using Step functions



Cloud Developers and Architects

- New and veteran Cloud developers
- Learn to harness the power AWS Lambda
- Create event-driven, serverless architecture
- Create workflows using Step functions

Additional Learning

- You must be able to understand [Python](#) code
- Watch Pluralsight's "AWS Developer" courses for more information on [S3](#), [API Gateway](#), [DynamoDB](#), [IAM](#), [CloudWatch](#) and other cloud services



Course Overview



Module 1: Getting Started with AWS Lambda

- Introducing AWS Lambda
- IAM 101

Module 2: Creating and Deploying Lambda Functions

- Writing, deploying, monitoring and troubleshooting

Module 3: Creating Serverless Architecture

- Creating event-driven architecture
- Using AWS services with Lambda functions

Module 4: Using Step Functions to Control Flow

- Coordinating AWS services using Step Functions
- Running and Monitoring Step Functions

Module 5: Deploying Serverless Applications

- Deploying AWS infrastructure
- Managing different lambda versions



AWS Lambda



Introduced in 2014

Execution time < 5min

Small, on-demand, event-driven applications

- File analysis
- IoT
- Backend services

Supported languages

- Node.js
- Python
- Java
- C#



Why Use AWS Lambda?



Simplify

Eliminate server
management
overhead (FaaS)



Why Use AWS Lambda?



Simplify

Eliminate server
management
overhead (FaaS)



Scale

automatically and
continuously



Why Use AWS Lambda?



Simplify

Eliminate server management overhead (FaaS)



Scale

automatically and continuously



Cost Effective

pay as you go,
only pay for
what you use



Why Use AWS Lambda?



Simplify

Eliminate server management overhead (FaaS)



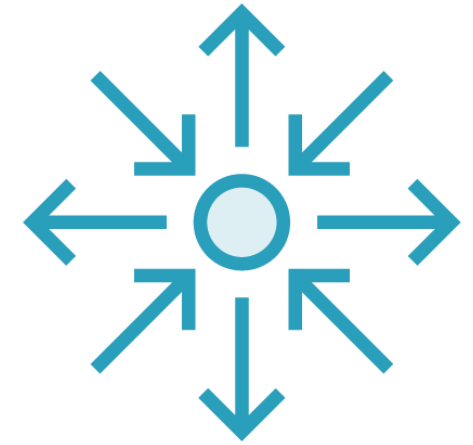
Scale

automatically and continuously



Cost Effective

pay as you go,
only pay for what you use



Always Ready

availability and
fault tolerant



AWS Lambda Costs

Requests

Free: first 1M

\$0.20 per 1M thereafter



AWS Lambda Costs

Requests

Free: first 1M

\$0.20 per 1M thereafter

Duration

Free: 400,000 GB/Sec

0.00001667 GB/Sec
thereafter



AWS Lambda Costs

Requests

Free: first 1M
\$0.20 per 1M thereafter

Duration

Free: 400,000 GB/Sec
0.00001667 GB/Sec
thereafter

Additional Charges

Data transfer
Other services
Lambda@Edge



AWS Lambda Limits

Resource	Limit
Maximum execution duration	300 sec (5 min)
Memory allocation	128 MB – 3008 MB
Ephemeral disk capacity	512 MB
Invoke request body size	6 MB (Sync) 128 K (Async)
Max Size of code/dependencies	250 MB
Concurrent executions	1000*



Getting Started with AWS Lambda



Sign up for AWS account

- Free tier

Learn about IAM

- Create users
- Policies



Demo



Creating your first *AWS* account

Logging to *AWS* console

Setting up new admin users



AWS Identity and Access Management



Users and Groups

Create users and assign
security credentials



AWS Identity and Access Management



Users and Groups

Create users and assign security credentials



Roles

Grants permissions to entities or AWS services



AWS Identity and Access Management



Users and Groups

Create users and assign security credentials



Roles

Grants permissions to entities or AWS services



Policies

Define permissions of an identity or resource



IAM Policies

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "dynamodb:UpdateItem"  
      ],  
      "Resource": "arn:aws:dynamodb:*:*:*"  
    }  
  ]  
}
```



Demo



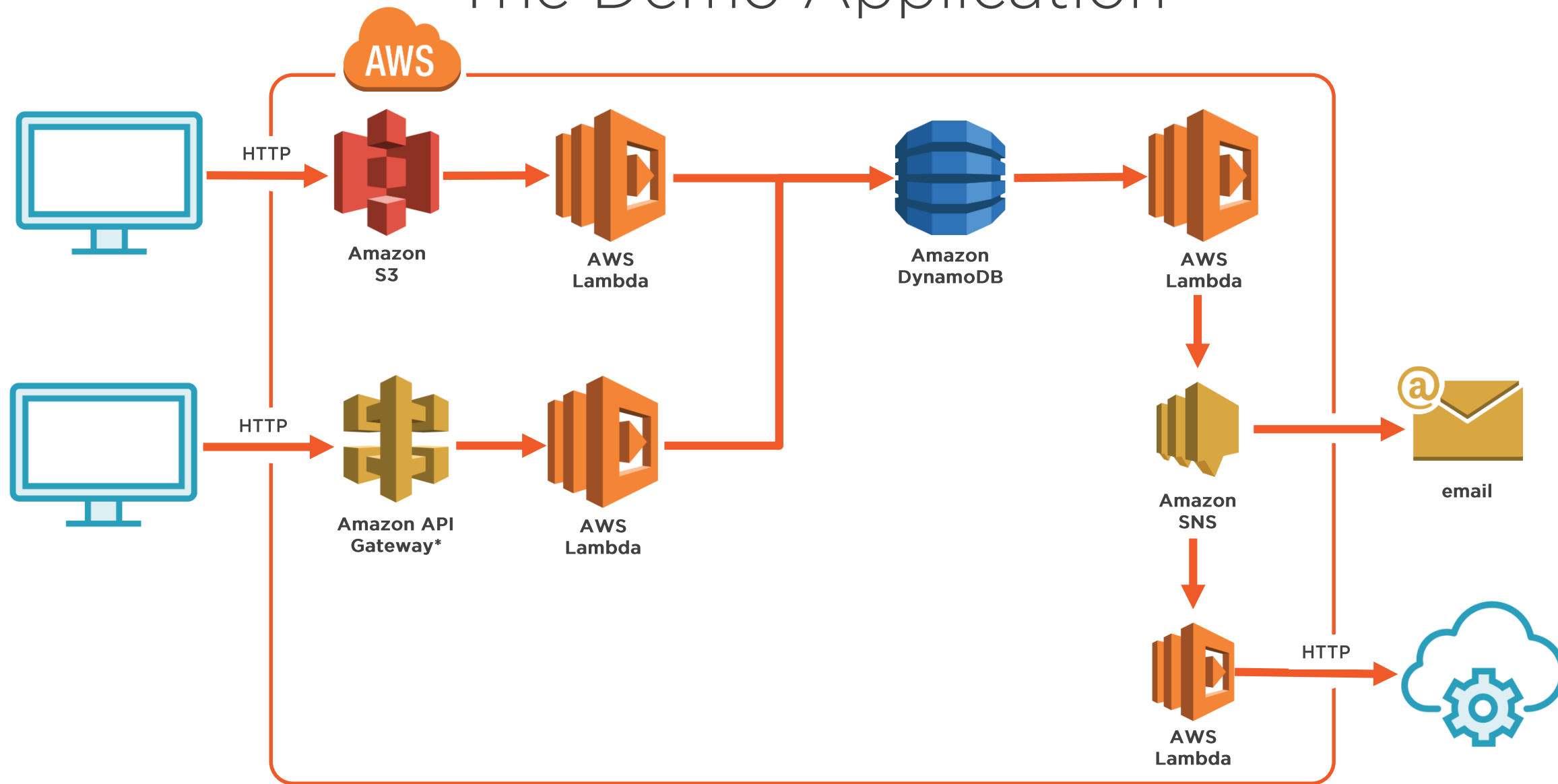
IAM Overview

Creating Roles

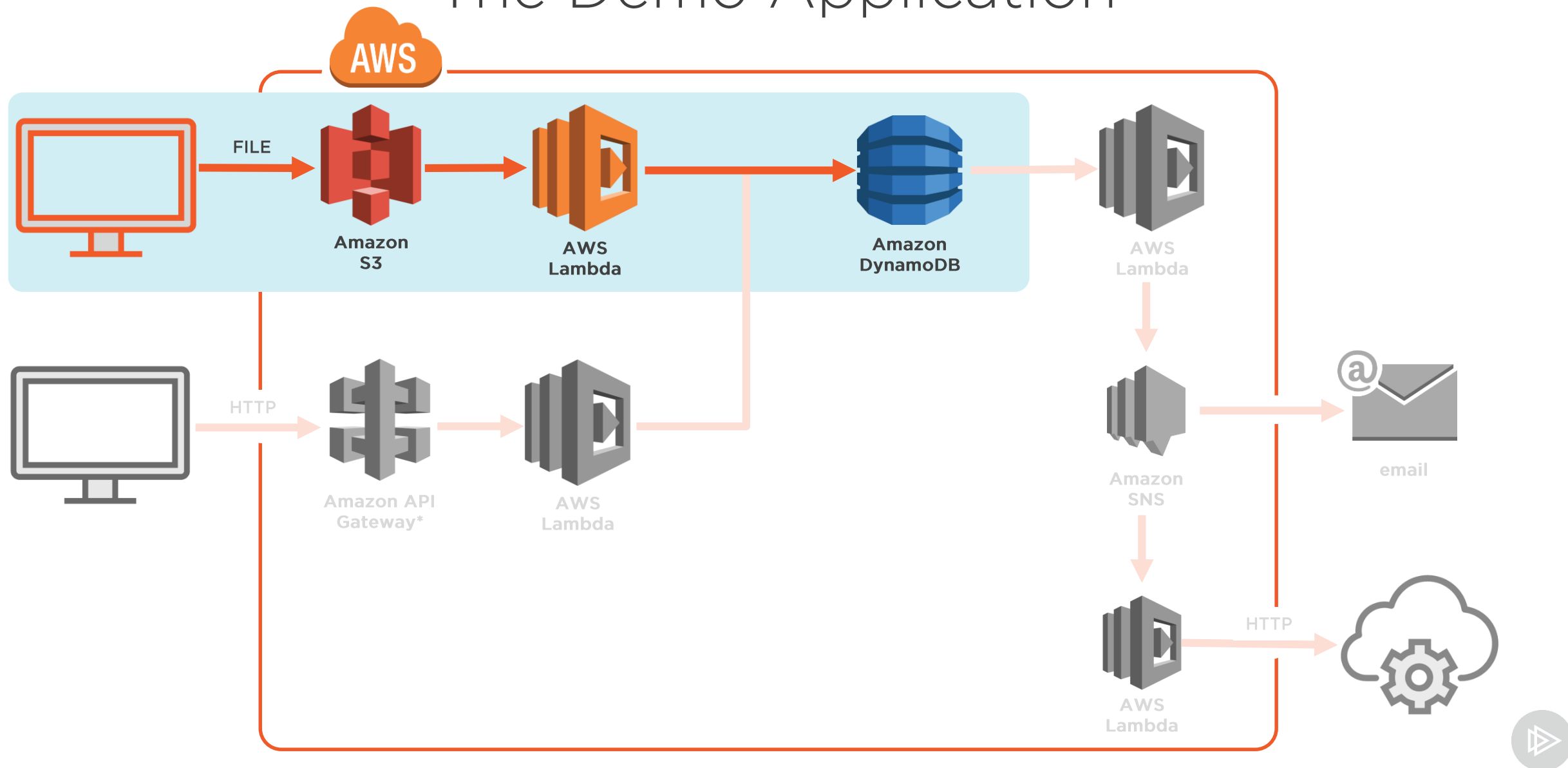
Managing Policies



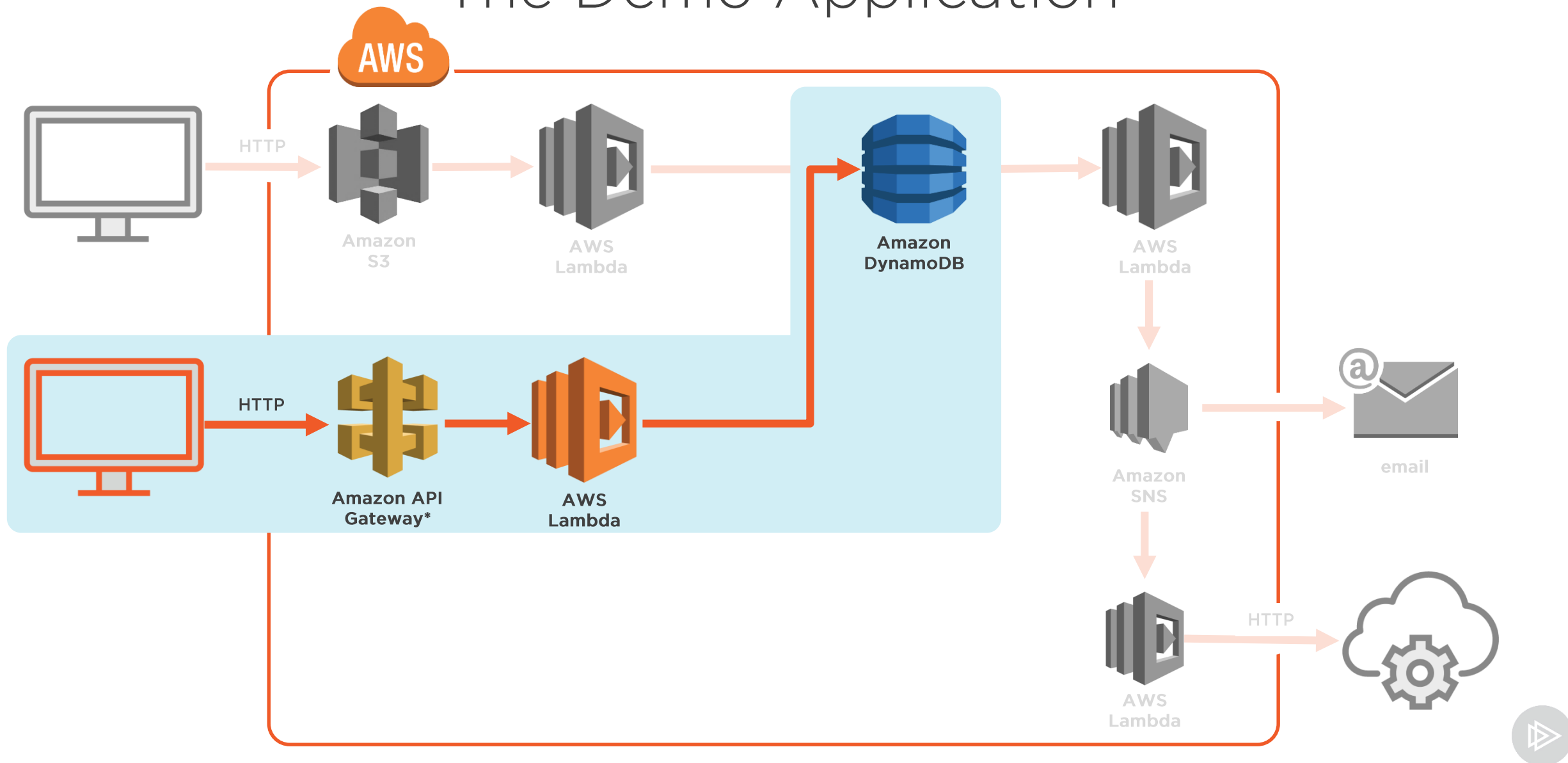
The Demo Application



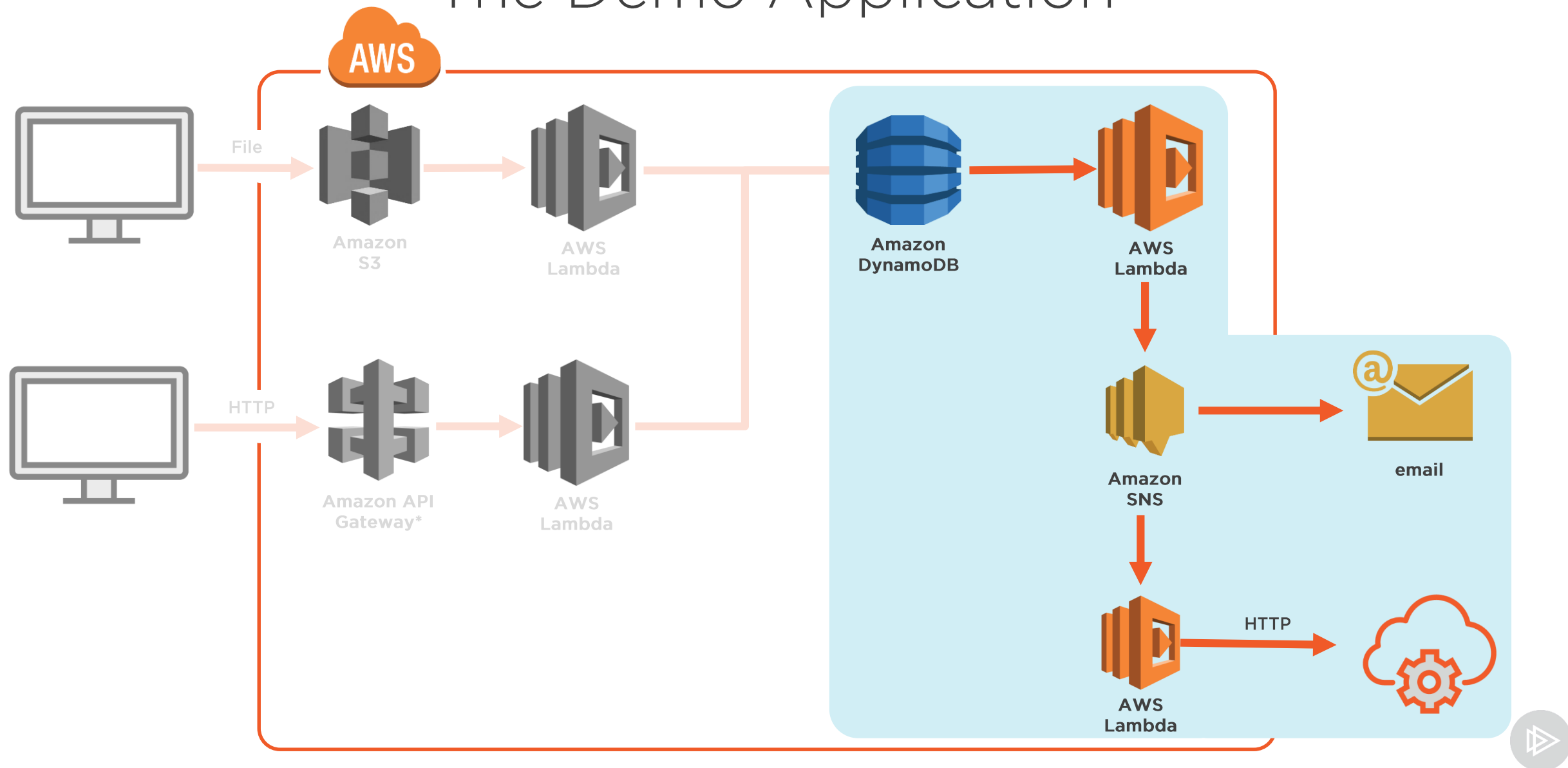
The Demo Application



The Demo Application



The Demo Application



Summary



Serverless introduction

AWS Lambda

Setting up AWS for the first time

IAM 101

