AWS re:Invent

GPSTEC348

.NET from the ground to the cloud

Ethan Ferdosi
Senior Solutions Architect – Microsoft Platform
Amazon Web Services

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

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

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Related sessions

TODAY

2:30 PM

- Developing serverless .NET Core on AWS (WIN308-R1)
- Build CI/CD pipeline for a .NET application in one hour (WIN402-R1)

3:15 PM

- Modernize legacy .NET applications on AWS (WIN304-R1)

TOMORROW

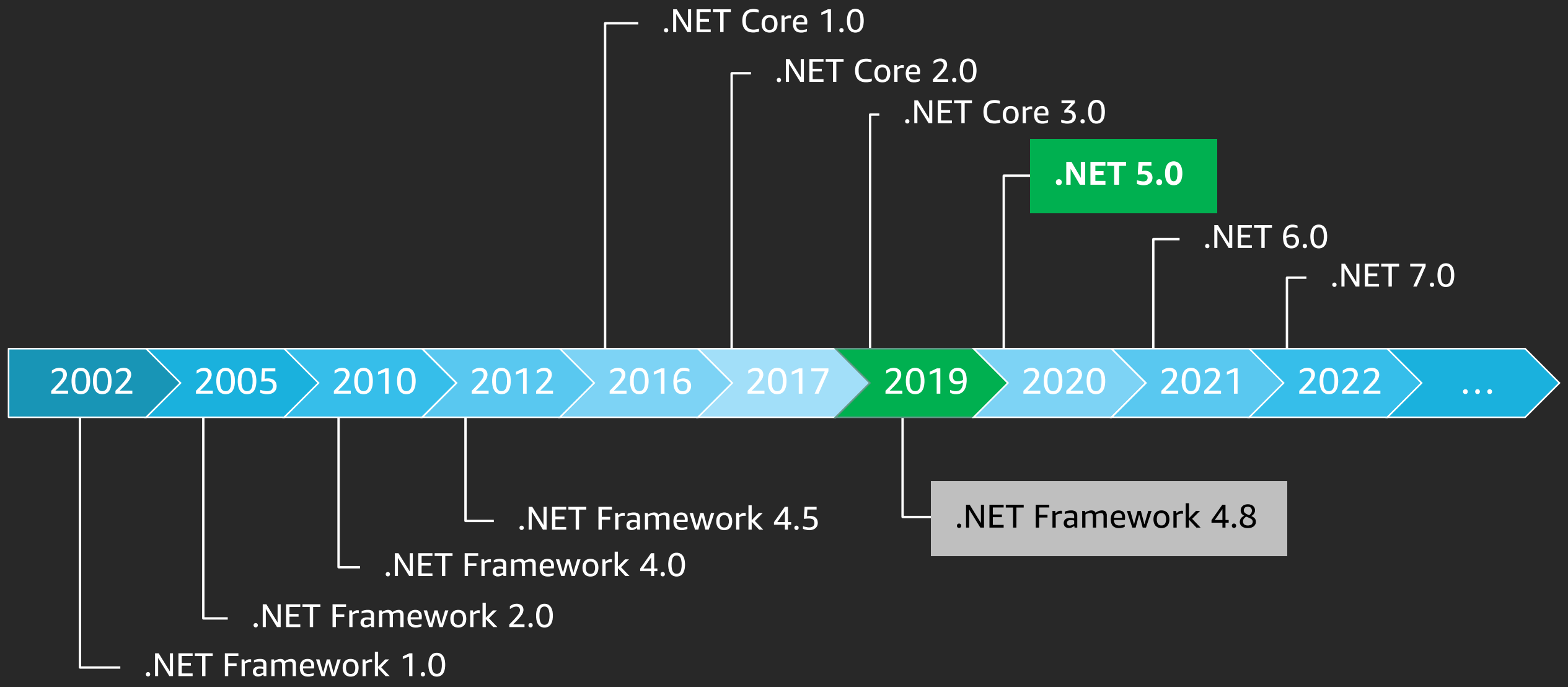
10:45 AM

- Converting a monolithic .NET app into a modern application (WIN315-R1)

What to expect

- Cloud migration strategies for .NET applications
- Benefits of modernizing .NET applications
- Running .NET Core on serverless
- Live demos!

Story of .NET


.NET Core 1.0
.NET Core 2.0
.NET Core 3.0
.NET 5.0
.NET 6.0
.NET 7.0
2002
2005
2010
2012
2016
2017
2019
2020
2021
2022
...
.NET Framework 4.5
.NET Framework 4.8
.NET Framework 4.0
.NET Framework 2.0
.NET Framework 1.0

Most common .NET framework apps



Windows & web service



ASP.NET Web Forms



MVC Web Application



Web API & microservices

Monolith

Cloud migration strategies







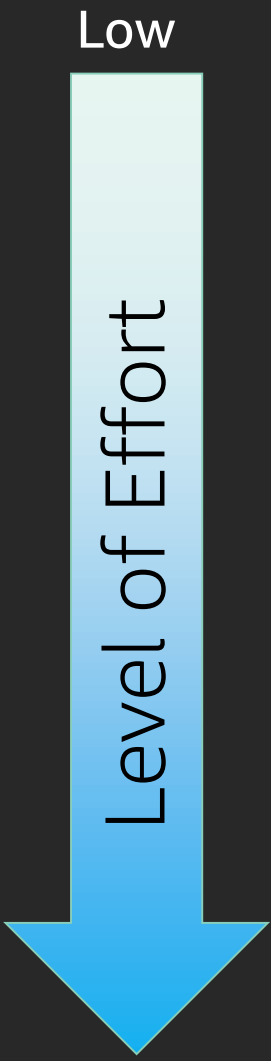




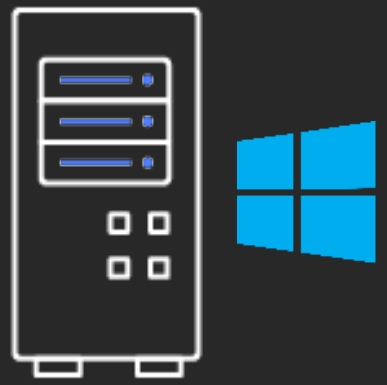


.NET Framework app migration strategies

Rehost	Deploy .NET Framework apps “as is” to Amazon EC2 virtual machines
Replatform	Containerize in Windows containers and orchestrate with Amazon ECS, AWS Fargate, or Amazon EKS
Refactor	Port to .NET Core, and run in Amazon ECS, Fargate, or Amazon EKS as Linux containers OR Lambda



Rehosting



Virtual Machine

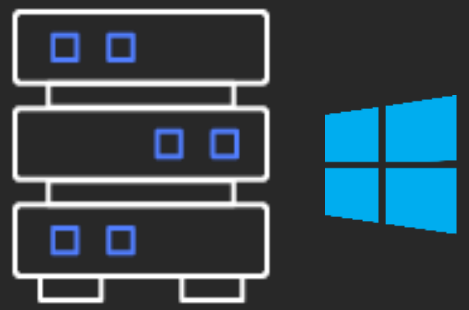
Pros

- Usually the fastest migration strategy
- No architectural modifications

Cons

- Miss most “cloudy” benefits of target platform—both technical and financial

Replatforming



Containers

Pros

- Realize some benefits of cloud platform
- Minimal architectural modifications

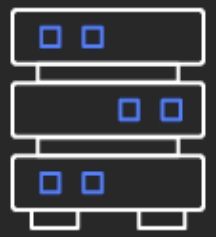
Cons

- Can be some rework required

Refactoring



Serverless



Linux Containers

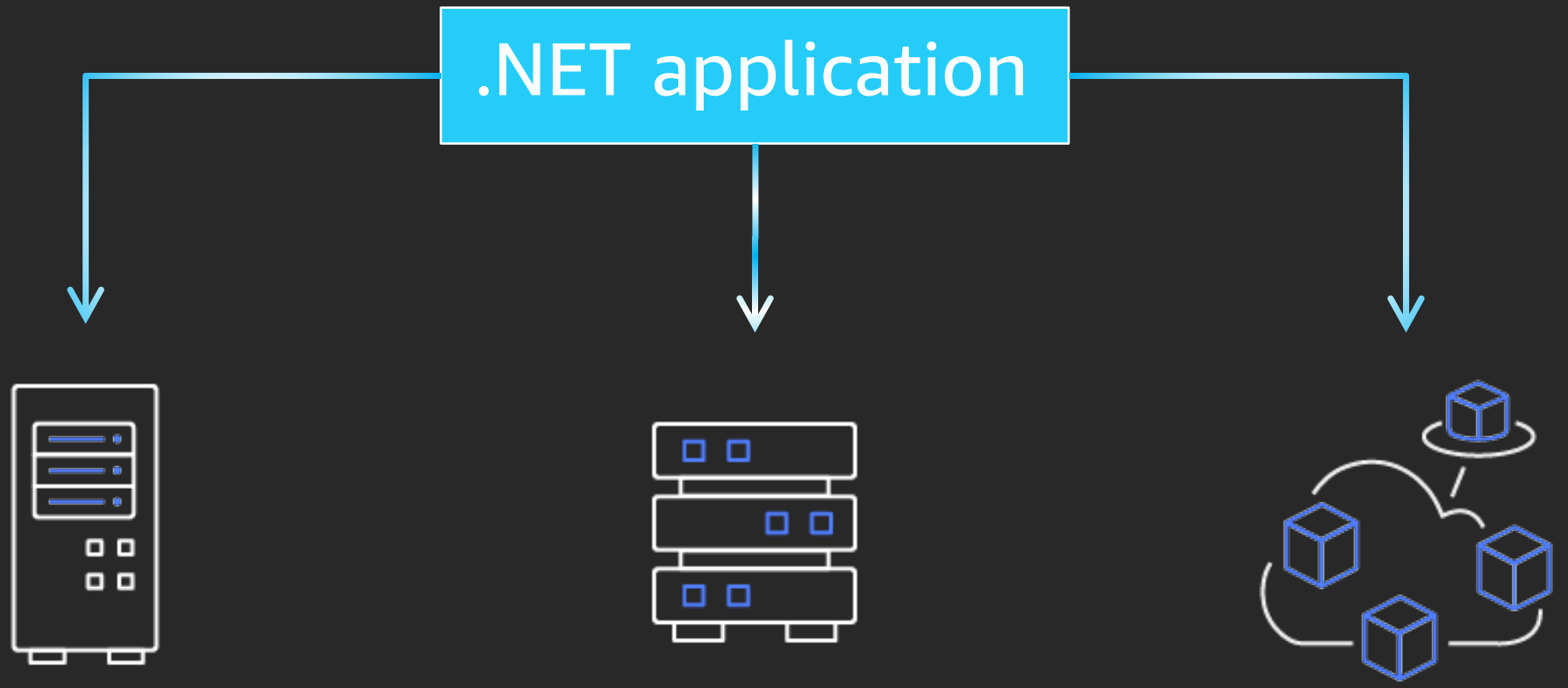
Pros

- Realize full potential of target cloud platform

Cons

- Requires the largest investment and level of effort

Hosting options for .NET applications



Virtual Machines
(Rehost)

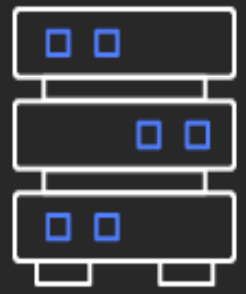
Containers
(Replatform)

Serverless
(Refactor)

Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Modernization paths for .Net applications



Linux
Containers



Serverless

Modernizing with serverless

Automatic scaling	Never pay for idle
No infrastructure provisioning	Highly available & fault tolerant

Serverless computing



AWS Lambda

Serverless event-driven code execution

Short-lived

All language runtimes

Data source integrations



Fargate

Serverless compute engine for containers

Long-running

Bring existing code

Fully managed orchestration

AWS Lambda

Event sources

HTTP Events

Via Amazon API Gateway

Service Events

Amazon S3, Amazon DynamoDB, etc.

Custom Events

Invoked by any of the AWS SDKs

Configuration

Memory (Indirectly CPU)

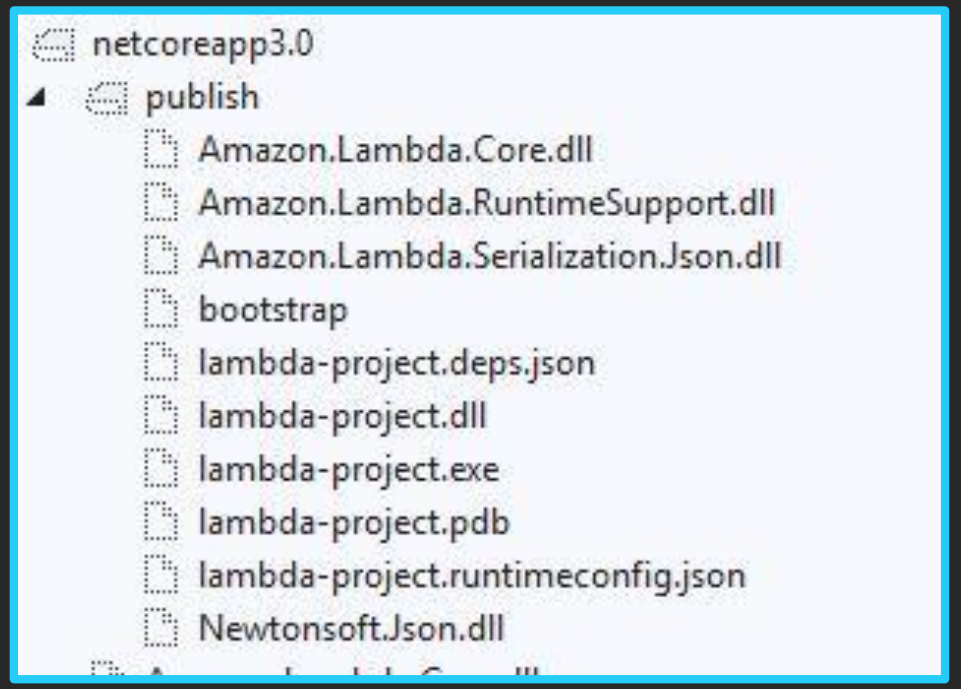
Timeout (Max 15 min)

Billing

Per 100 ms per function execution time

.NET Core Lambda function

- .NET Core project (targeting netcoreapp3.0)
- Packaged using dotnet publish



Recommended AWS tools

- AWS Toolkit for Visual Studio
- Amazon.Lambda.Tools— Extension to dotnet CLI

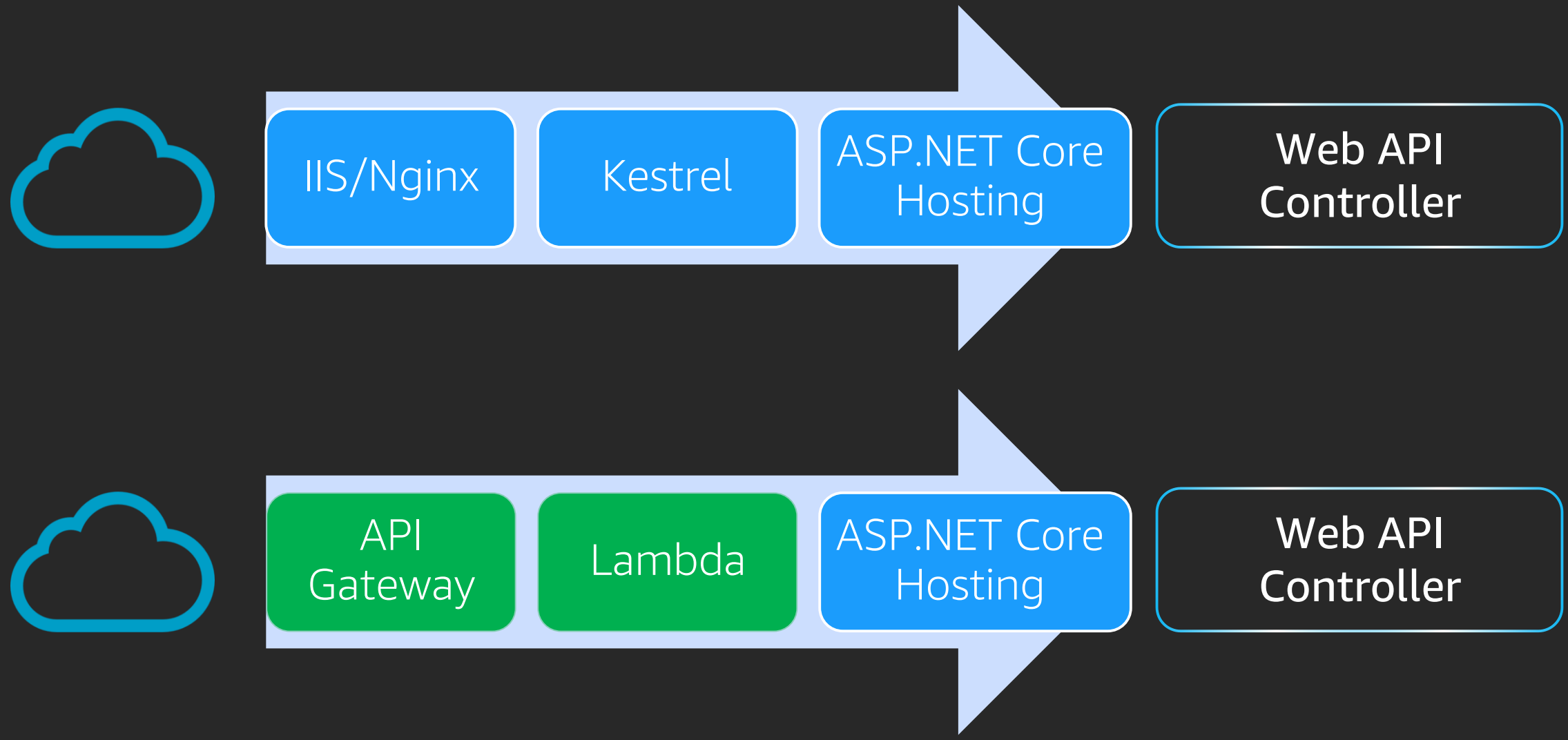
.NET Core Lambda function entry

```
public async Task FunctionHandler(S3Event input, ILambdaContext context)
{
    //TODO: Your code goes here...
    return;
}
```

Function handler: <assembly>::<full type name>::<method>

```
"function-runtime" : "dotnetcore2.1",
"function-memory-size" : 256,
"function-timeout" : 30,
"function-handler" : "AWSLambda4::AWSLambda4.Function::FunctionHandler"
}
```

ASP.NET Core web application on Lambda



ASP.NET Core web application on Lambda

Nuget package **Amazon.Lambda.AspNetCoreServer**

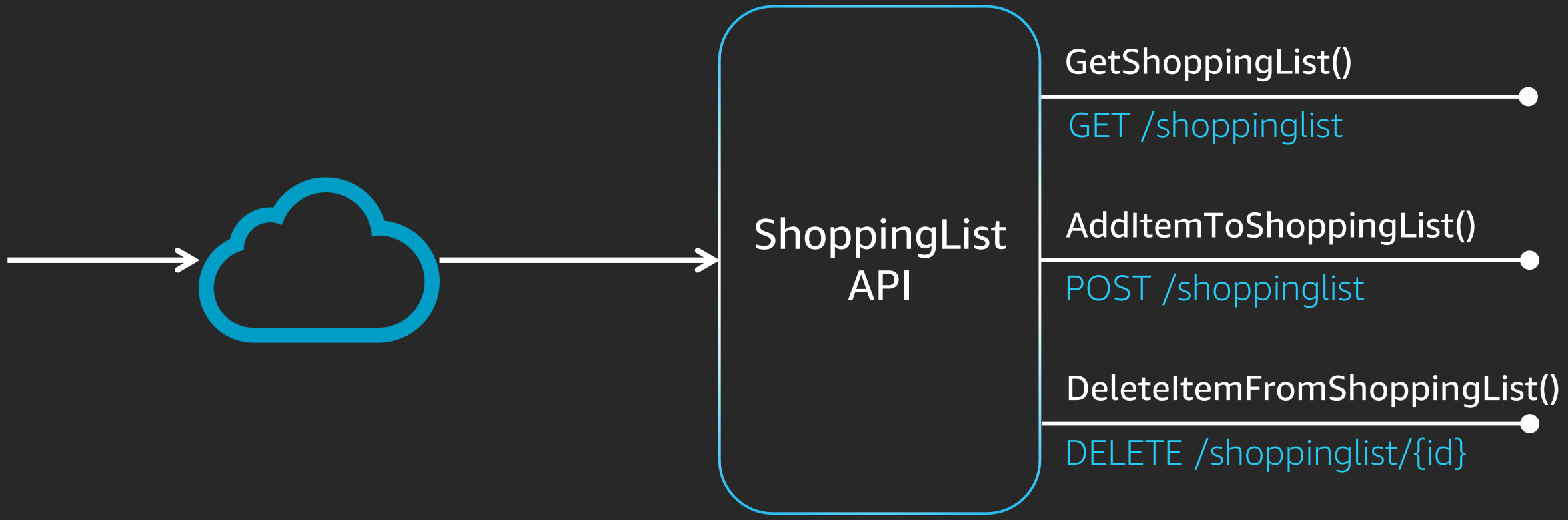
Pros

- Easy to migrate existing applications
- Little to learn about serverless

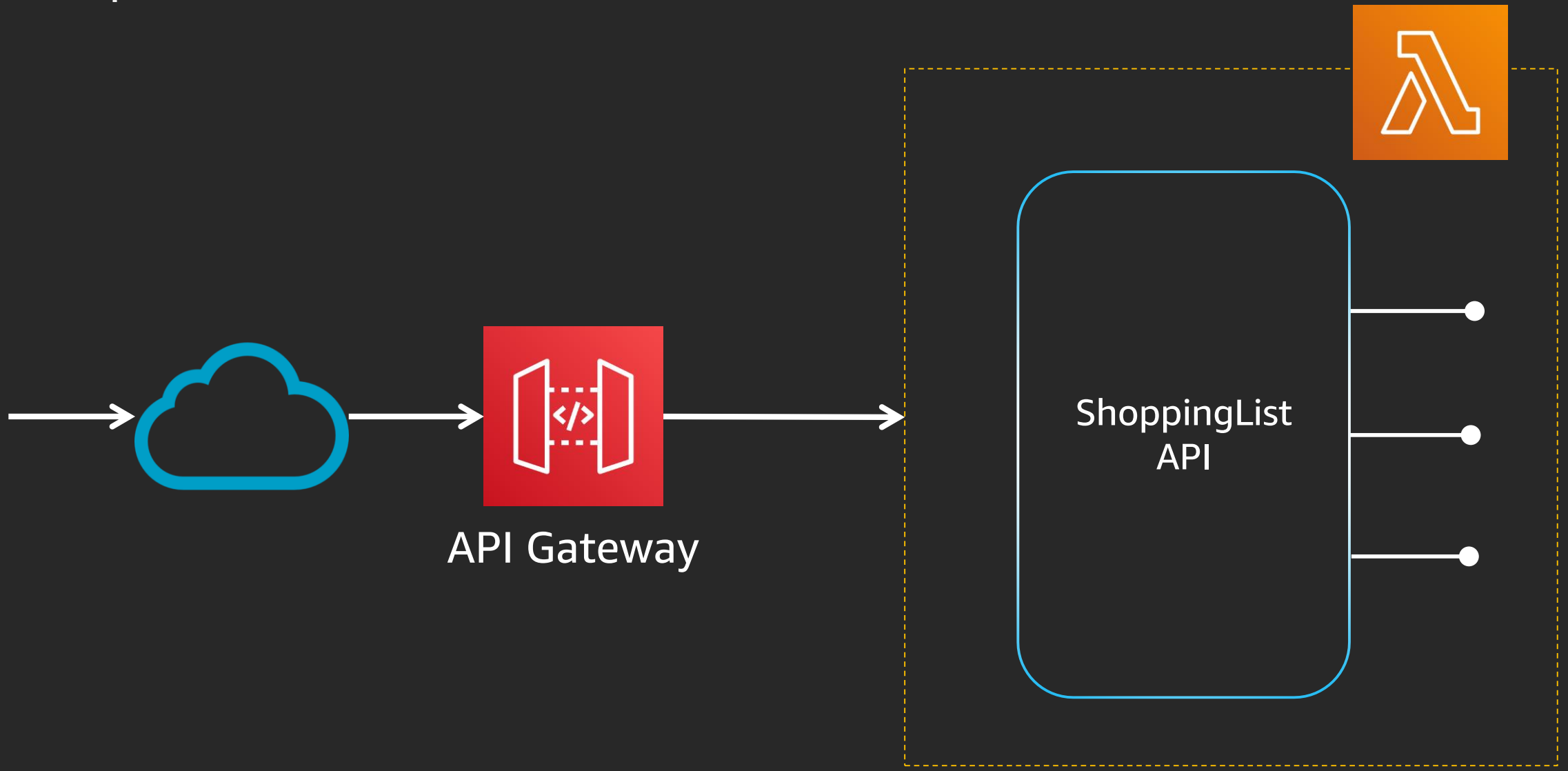
Cons

- Larger package bundle
- Cost to initialize ASP.NET Core framework

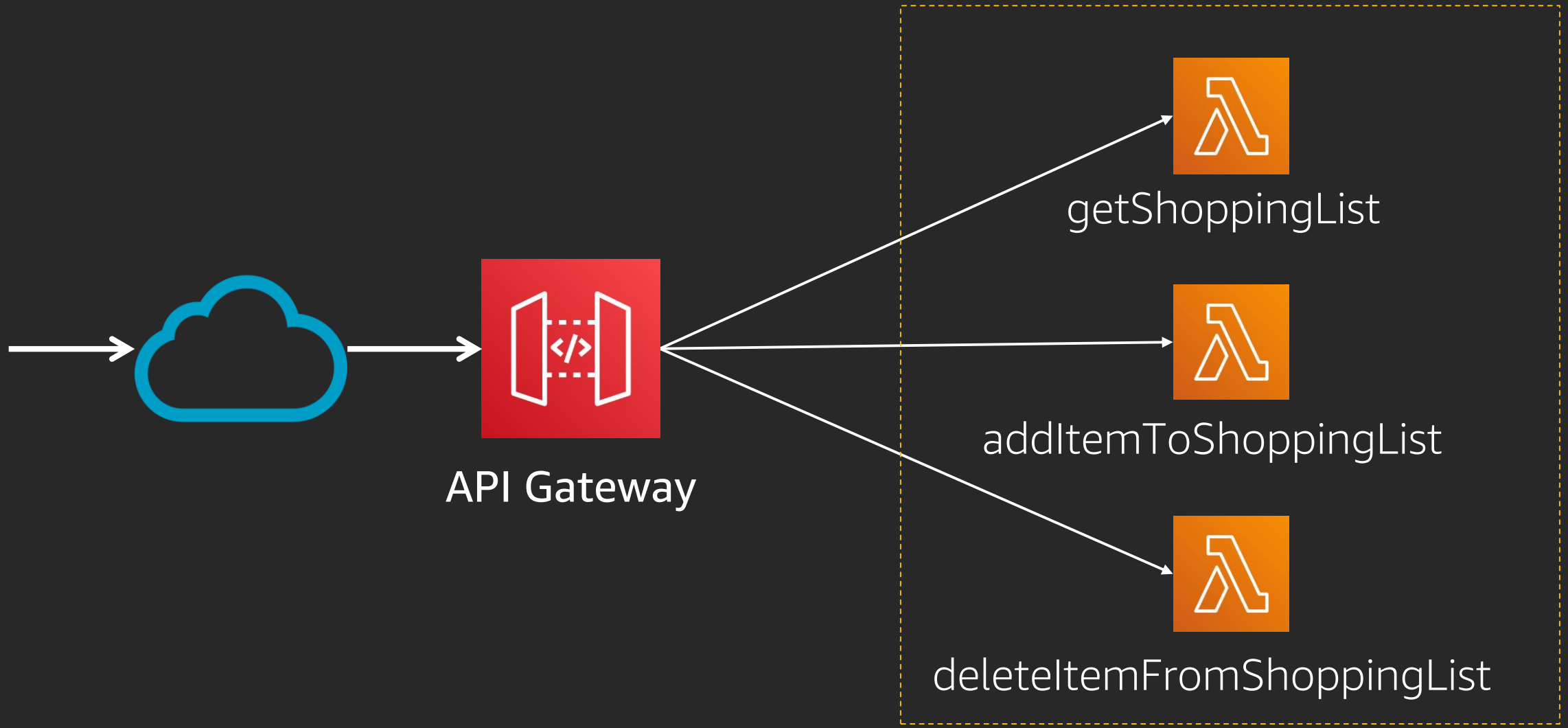
Running .NET Core Web API on Lambda



Option 1: Web API to Lambda service



Option 2: Web API to Lambda functions



Lambda function logging



- Console class

```
Console.WriteLine("<your log…>");
```

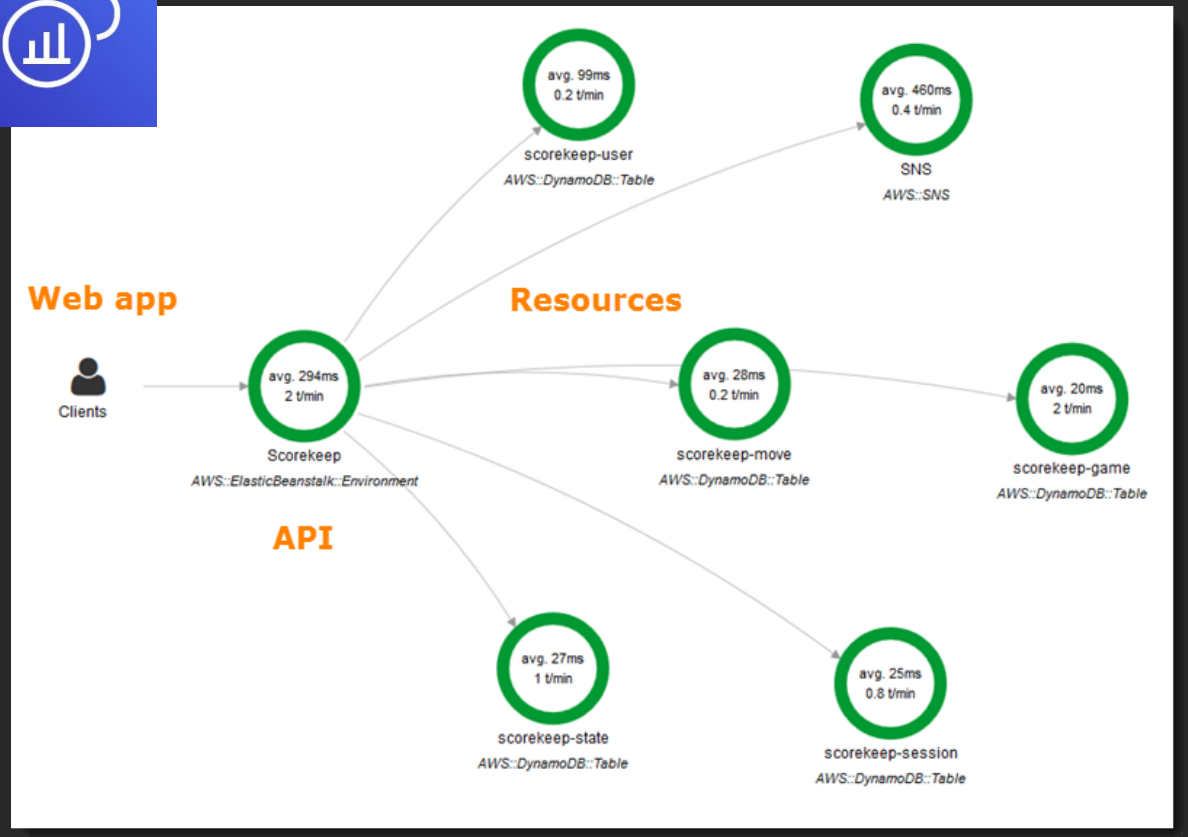
- LambdaLogger class

```
LambdaLogger.Log("<your log…>");
```

- Context object

```
Context.Logger.Log("<your log…>");
```

Tracing .NET Core app with AWS X-Ray



- Logs request and responses
- Calls to downstream resources, microservices, databases and HTTP web APIs
- Nuget packages

AWSXRayRecorder.*

Demo

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Debugging .NET Core Lambda



AWS Cloud9



AWS Toolkit for Visual Studio Code



Serverless Application Model (SAM)



JetBrains Rider

Why .NET on AWS?

AWS provides full support for .NET applications and Windows workloads

AWS services support the latest release of .NET Framework and .NET Core

AWS provides deep integration with Visual Studio and Visual Studio Team Services

AWS provides a range of resources and tools to build modern .NET serverless solutions

APN Navigate for Microsoft Workloads—call to action



Microsoft Workloads

APN Partner's path to specializing in Microsoft workloads on AWS

Technical enablement including whitepapers, blogs, and architecture designs

Business enablement such as first call decks and containers sales kit templates

Key outputs: Case studies, microsite, public customer references

Get started today!

Thank you!

Ethan Ferdosi

eferdosi@amazon.com

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.