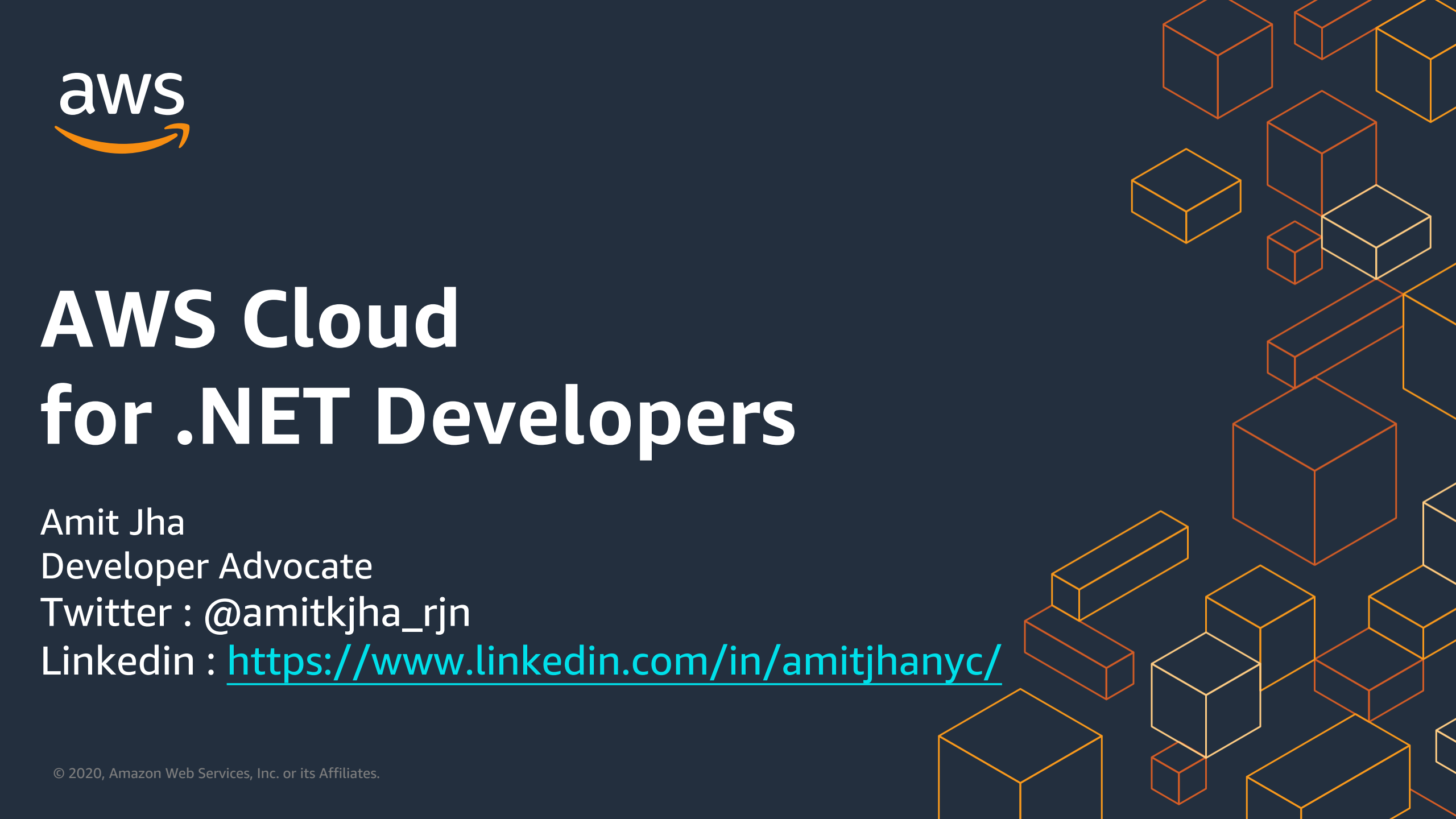


AWS Cloud for .NET Developers

Amit Jha
Developer Advocate
Twitter : @amitkjha_rjn
Linkedin : https://www.linkedin.com/in/amitjhanyc/

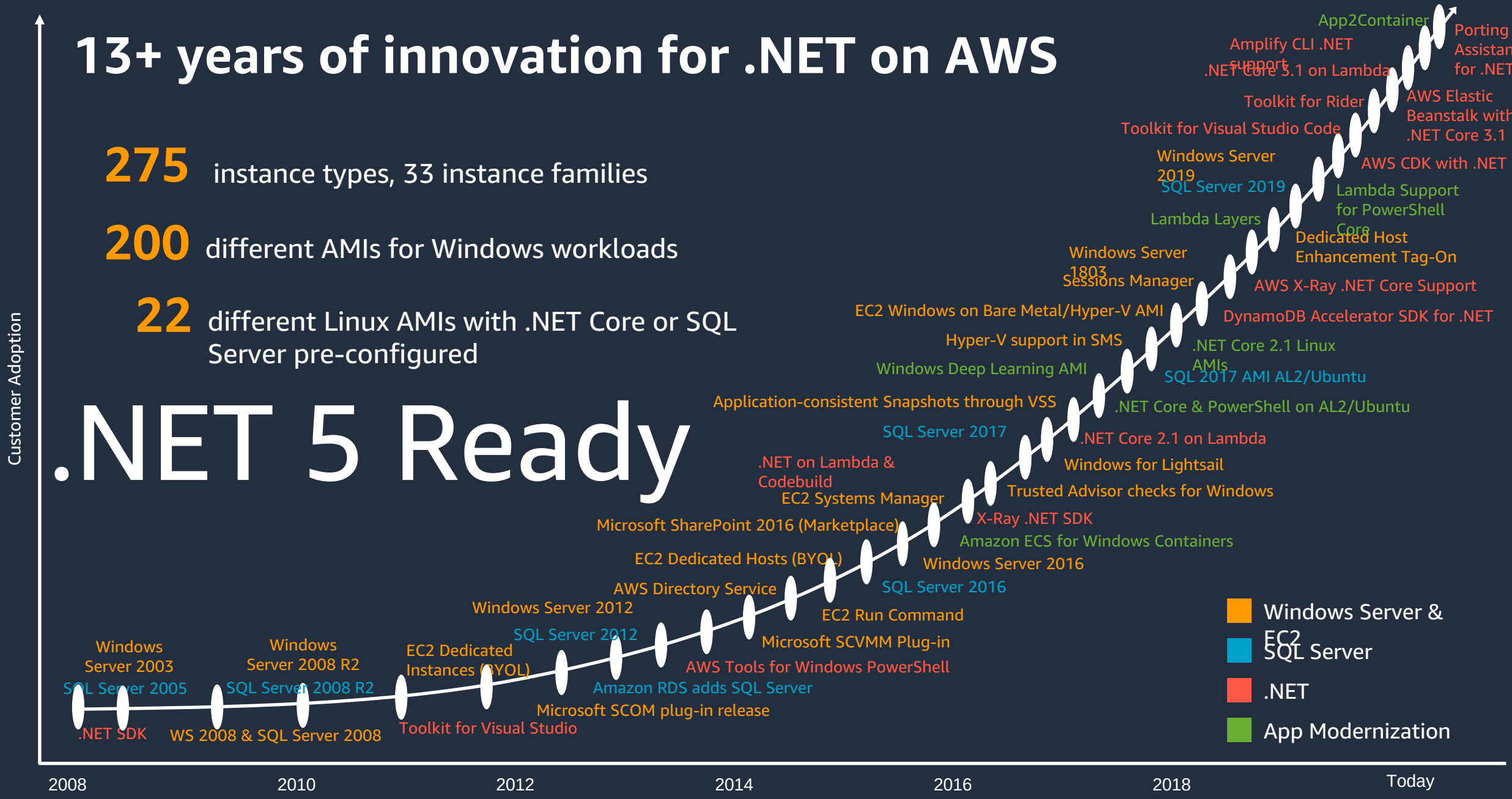
13+ years of innovation for .NET on AWS

275 instance types, 33 instance families

200 different AMIs for Windows workloads

22 different Linux AMIs with .NET Core or SQL Server pre-configured

.NET 5 Ready



AWS Tools for .NET Developers

aws

Development Tools

IDE integration

- AWS Toolkit for Visual Studio
- AWS Toolkit for Visual Studio Code
- AWS Toolkit for Rider

Programmable SDK

- AWS SDK for .NET
- AWS CDK for .NET

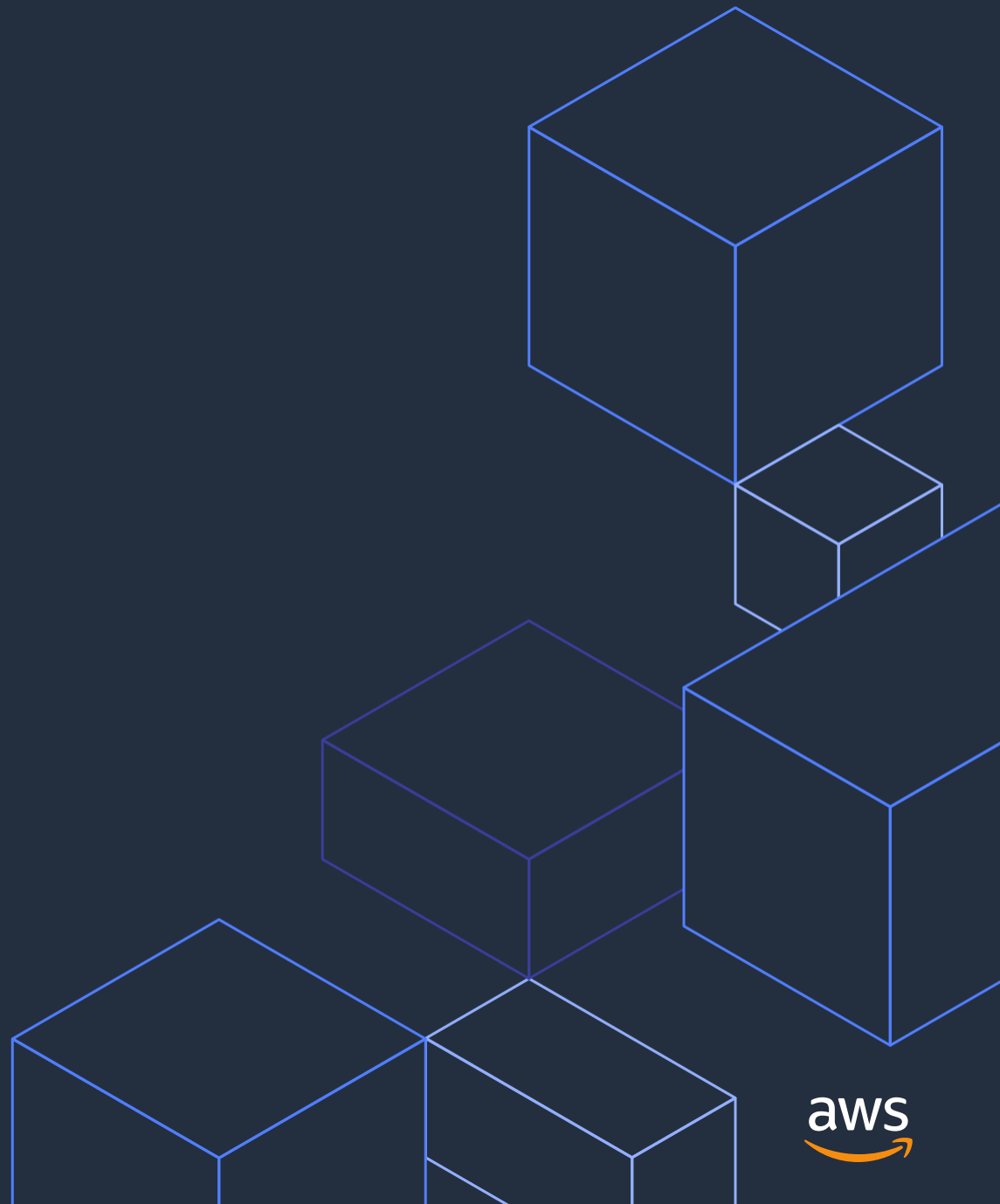
Command line tools

- AWS Tools for PowerShell
- ‘dotnet’ CLI extensions
- AWS CLI
- AWS SAM for Windows

CI/CD integration

- AWS Tools for Azure DevOps
- AWS CodePipline/ CodeBuild

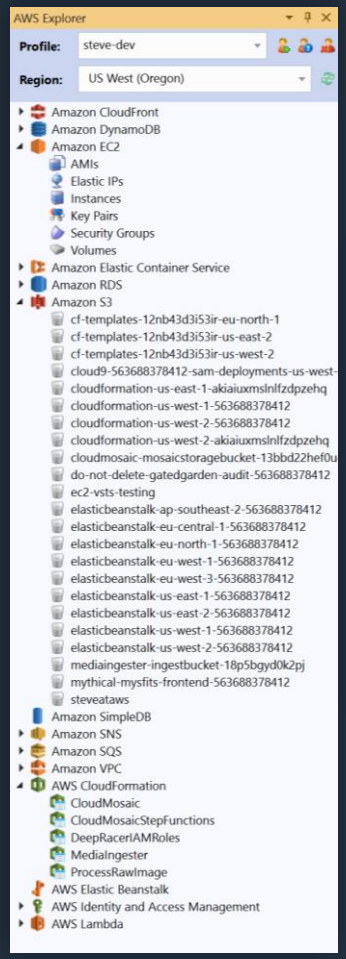
IDE Integration

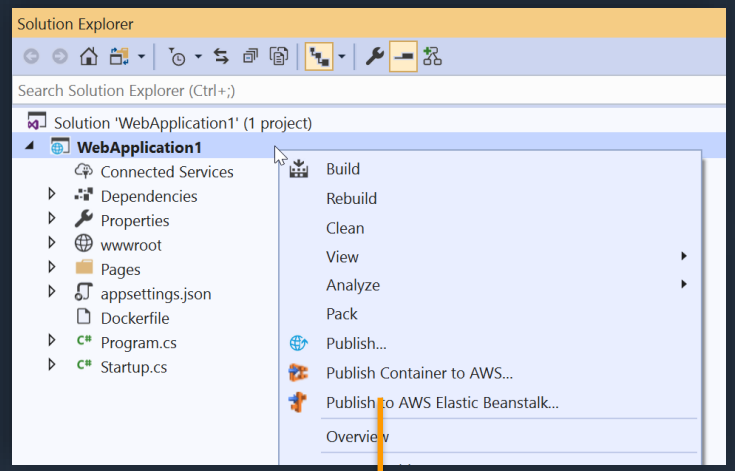


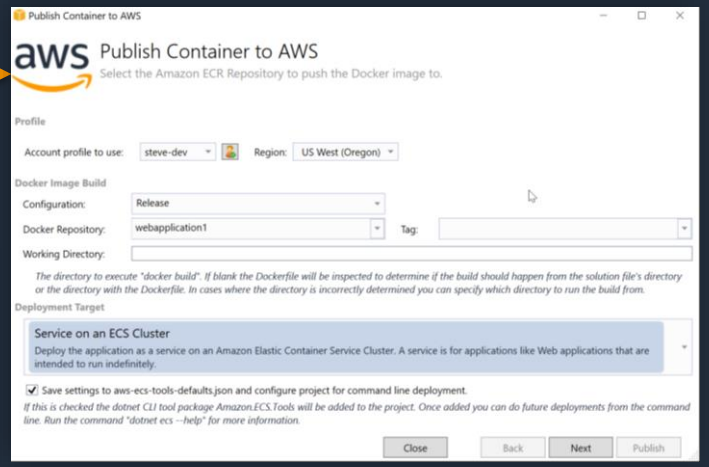
aws

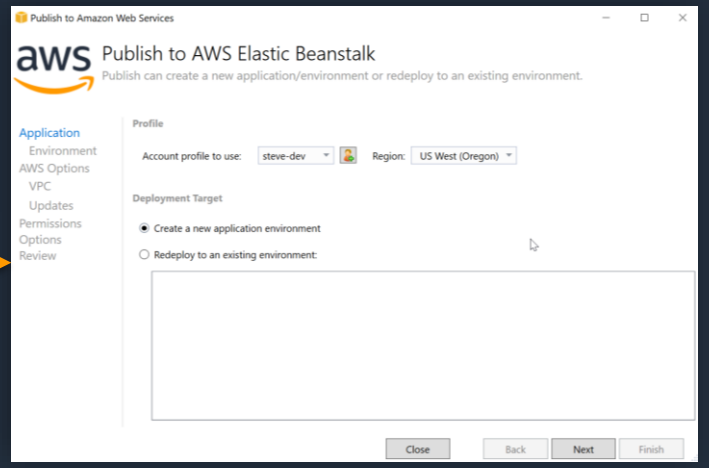
AWS Toolkit for Visual Studio

AWS Toolkit for Visual Studio



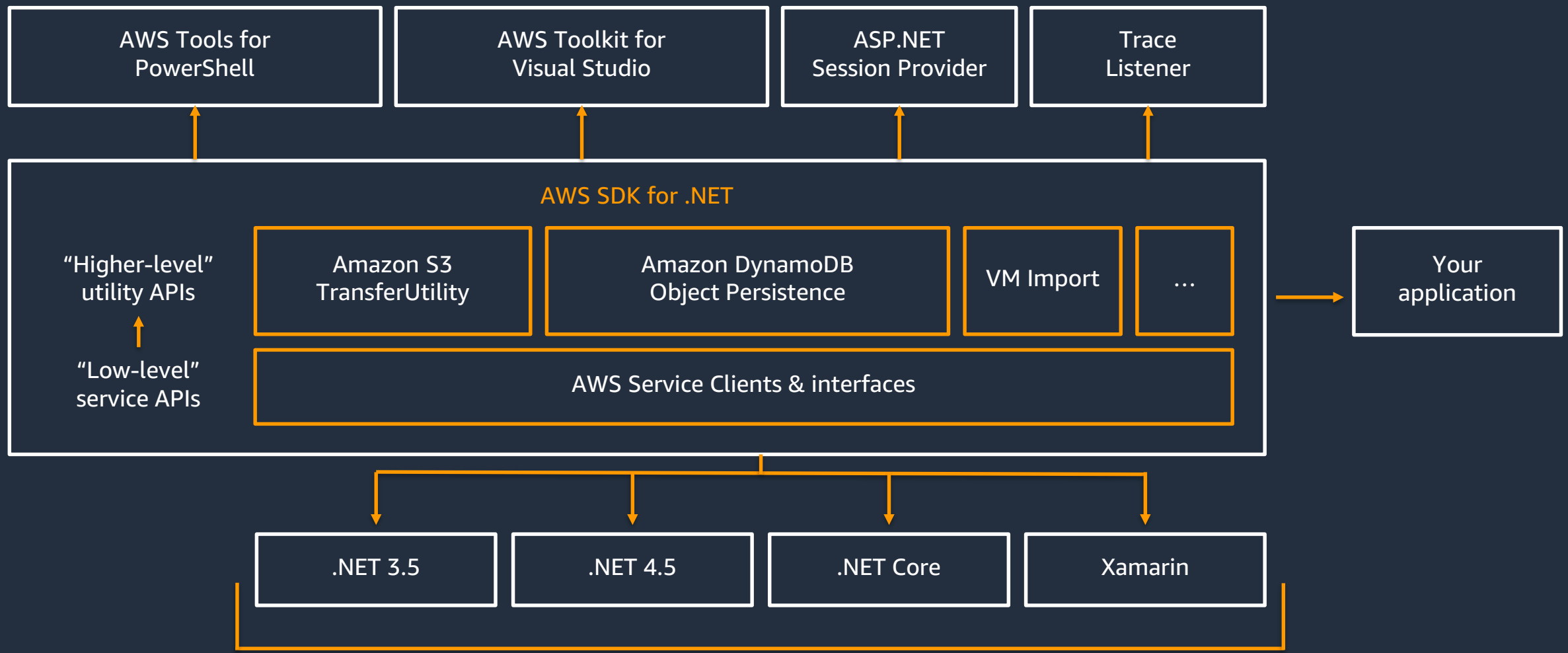






"Publish to AWS Lambda" wizard also available (not shown)

Core Enabling Technology



AWS SDK for .NET

.NET .NET Core

NuGet packages targeting .NET Standard 1.3 and 2.0

aws

AWS Cloud : Breadth and Depth of Services

More services and more functionality within those services

Consistent SDK Coding Pattern


```
using servicename;
using servicename.Model;

var client = new AmazonservicenameClient();
operationnameResponse response = await client.operationnameAsync(new operationnameRequest
{
    RequestProperty1 = 'some data',
    RequestProperty2 = new List<string> { 'someother', 'data' }
});

foreach (var element in response.OutputDataMember)
{
   // do something
}
```


aws

AWS Tools for PowerShell

Available for Windows PowerShell and PowerShell 6+

Use to manage AWS resources and services

Over 5,000 cmdlets across 150+ services

Distributed on PowerShell Gallery

AWSPowerShell

- Windows PowerShell v2-v5.1
- Pre-installed on Amazon-provided EC2 Windows images

AWSPowerShell.NetCore

- Windows PowerShell v3-5.1
- PowerShell 6+ (Windows, macOS & Linux)

If you can code something in an SDK, you can script it in PowerShell

aws

‘dotnet’ CLI extensions

Administrator: Windows PowerShell

```
PS D:\> dotnet new -l
Usage: new [options]

Options:
  -h, --help          Displays help for this command.
  -l, --list          Lists templates containing the specified name. If no name is specified, lists all templates.
  -n, --name          The name for the output being created. If no name is specified, the name of the current directory is used.
  -o, --output        Location to place the generated output.
  -i, --install       Installs a source or a template pack.
  -u, --uninstall     Uninstalls a source or a template pack.
  --nuget-source      Specifies a NuGet source to use during install.
  --type              Filters templates based on available types. Predefined values are "project", "item" or "other".
  --force             Forces content to be generated even if it would change existing files.
  -lang, --language   Filters templates based on language and specifies the language of the template to create.


Templates                                           Short Name                          Language          Tags
--------------------------------------------------------------------------------------------------------------------------------------
Order Flowers Chatbot Tutorial                      lambda.OrderFlowersChatbot          [C#]              AWS/Lambda/Function
Lambda Detect Image Labels                          lambda.DetectImageLabels            [C#], F#          AWS/Lambda/Function
Lambda Empty Function                               lambda.EmptyFunction                [C#], F#          AWS/Lambda/Function
Lex Book Trip Sample                                lambda.LexBookTripSample            [C#]              AWS/Lambda/Function
Lambda Simple DynamoDB Function                     lambda.DynamoDB                     [C#], F#          AWS/Lambda/Function
Lambda Simple Kinesis Firehose Function             lambda.KinesisFirehose              [C#]              AWS/Lambda/Function
Lambda Simple Kinesis Function                      lambda.Kinesis                      [C#], F#          AWS/Lambda/Function
Lambda Simple S3 Function                           lambda.S3                           [C#], F#          AWS/Lambda/Function
Lambda Simple SQS Function                          lambda.SQS                          [C#]              AWS/Lambda/Function
Lambda ASP.NET Core Web API                         serverless.AspNetCoreWebAPI         [C#], F#          AWS/Lambda/Serverless
Lambda ASP.NET Core Web Application with Razor Pages serverless.AspNetCoreWebApp        [C#]              AWS/Lambda/Serverless
Serverless Detect Image Labels                      serverless.DetectImageLabels        [C#], F#          AWS/Lambda/Serverless
Lambda DynamoDB Blog API                            serverless.DynamoDBBlogAPI          [C#]              AWS/Lambda/Serverless
Lambda Empty Serverless                             serverless.EmptyServerless          [C#], F#          AWS/Lambda/Serverless
Lambda Giraffe Web App                              serverless.Giraffe                  F#                AWS/Lambda/Serverless
Serverless Simple S3 Function                       serverless.S3                       [C#], F#          AWS/Lambda/Serverless
Step Functions Hello World                          serverless.StepFunctionsHelloWorld  [C#], F#          AWS/Lambda/Serverless
```


aws

AWS Toolkit for Visual Studio Code

Open source plug-in for the Rider IDE that makes it easier to create, debug, and deploy .NET applications on Amazon Web Services:

"Getting started" project template

Step-through debugging

Deployment from the IDE

https://aws.amazon.com/rider/
https://github.com/aws/aws-toolkit-jetbrains

aws

AWS Toolkit for Rider

Integrated experience targeting development of serverless applications in Node.js, Python and .NET:

Select a quickstart serverless application template.

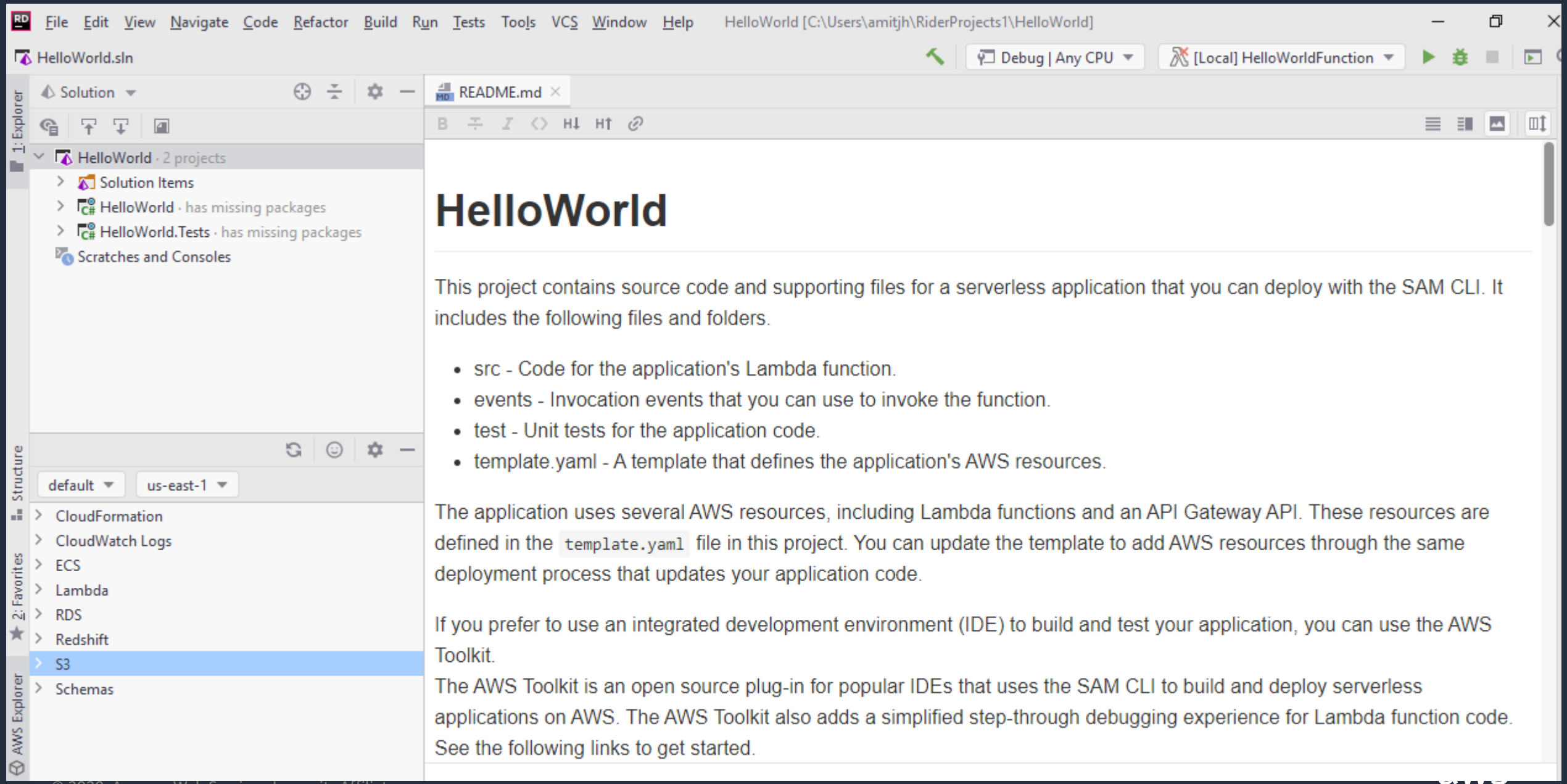
Step-through debugging

Deployment from the IDE

Access CloudWatch From the IDE

https://aws.amazon.com/visualstudiocode/
https://github.com/aws/aws-toolkit-vscode

aws



Compute and Operations

More

Opinionated

Less

	AWS manages	Customer manages	
AWS Lambda Serverless functions	Data source integrations Physical hardware, software, networking, and facilities Provisioning	Application code	
AWS Fargate Serverless containers	Container orchestration, provisioning Cluster scaling Physical hardware, host OS/kernel, networking, and facilities	Application code Data source integrations Security config and updates Network config	Management tasks
Amazon ECS/EKS Container-management -as-a-service	Container orchestration control plane Physical hardware software, networking, and facilities	Application code Data source integrations Work clusters	Security config and updates, network config, firewall, management tasks
Amazon EC2 Infrastructure-as-a-Service	Physical hardware software, networking, and facilities	Application code Data source integrations Scaling Security config and updates Network config Management tasks	Provisioning, managing scaling and patching of servers

AWS Elastic Beanstalk

A fully managed service for hosting web applications

aws

Low Management Overhead

Automated App Deployments

- Safely deploys app updates while monitoring app health
- Choose from all-at-once, rolling, traffic splitting or blue/green deployments
- Durably stores each app version to S3 for easy rollback

Automated Platform Updates

- Keeps the operating system and runtime updated with latest patches and updates
- Reduces update costs by $100/server/yr. vs. manually applied updates
- Monthly releases of new platform updates
- Configurable update schedule

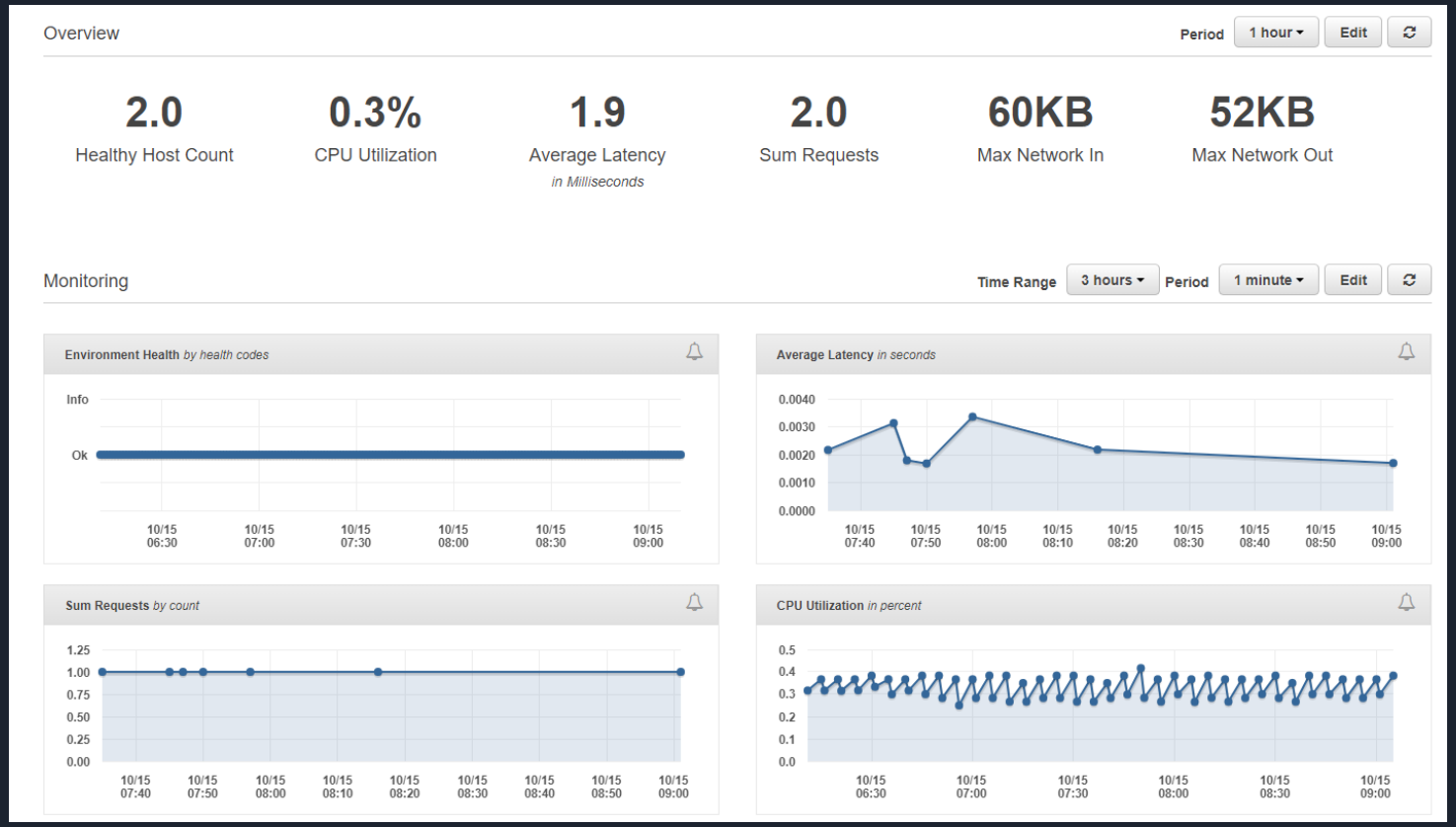
App Health Monitoring

- Summarizes app health based on multiple inputs
- Monitors HTTP error rates & request/response latency
- Monitors resource usage (CPU, network, memory)
- Logs health events
- Sends notifications based on user-defined thresholds

aws

Stay appraised of application health

- Dashboard with detailed application metrics
- Log events with CloudWatch Logs streaming
- Easily enable AWS X-Ray for analyzing and debugging



Low management overhead

aws

The AWS Machine Learning Stack

Broadest and most complete set of ML capabilities

AI SERVICES

VISION
- Amazon Rekognition

SPEECH
- Amazon Polly
- Amazon Transcribe *+Medical*

TEXT
- Amazon Comprehend *+Medical*
- Amazon Translate
- Amazon Textract

SEARCH
- Amazon Kendra

CHATBOTS
- Amazon Lex

PERSONALIZATION
- Amazon Personalize

FORECASTING
- Amazon Forecast

FRAUD
- Amazon Fraud Detector

DEVELOPMENT
- Amazon CodeGuru

CONTACT CENTERS
- Contact Lens *For Amazon Connect*

ML SERVICES

Amazon SageMaker

SageMaker Studio IDE

Ground Truth | AWS Marketplace for ML | Built-in algorithms | Notebooks | Experiments | Processing | Model training & tuning | Debugger | Autopilot | Model hosting | Model Monitor | Neo | Augmented AI

ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow | mxnet | PYTORCH | GLUON | Keras | DeepGraphLibrary

Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Inferentia | FPGA

aws

e.g. AWS AI Services : Pre-trained / Auto trained models

- Sentiment Analysis and NLP: Amazon Comprehend
- Image and Video Analysis: Amazon Rekognition
- Document Translation: Amazon Translate
- Speech to Text: Amazon Transcribe
- Text to Speech: Amazon Polly
- Chatbots: Amazon Lex

aws

AWS SDK for .NET – Common Pattern for Calling Services

```
// Create Service Client
var serviceClient = new AmazonServiceClient(Amazon.RegionEndpoint.SomeRegion);

var serviceRequest = new ServiceRequest
{
    RequestProperty1 = value1,
    RequestProperty2 = value2,
};

// Call ServiceMethod to convert the text
var serviceResponse = await serviceClient.ServiceMethodAsync(serviceRequest);

// Process the serviceResponse
```


aws

Amazon Comprehend

Discover insights and relationships in text

Comprehensive set of enterprise-grade tools

CI/CD Tools


AWS CodeStar


AWS CodeBuild


AWS CodeCommit


AWS CodeDeploy

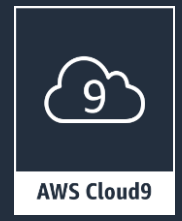

AWS CodePipeline

Infrastructure as Code


AWS CloudFormation

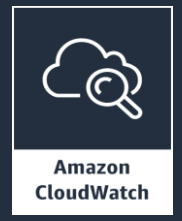

AWS Cloud Dev. Kit (CDK)

IDE

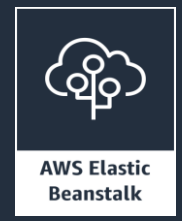

AWS Cloud9

Monitoring & Tracing


AWS X-Ray


Amazon CloudWatch

Web Apps


AWS Elastic Beanstalk

IDE and DevOps Toolkits


Visual Studio Code


IntelliJ


PyCharm


Visual Studio


Eclipse

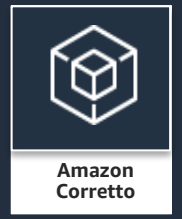

VSTS

CLI and Scripting Tools


AWS CLI


Tools for PowerShell

Languages


Amazon Corretto

Mobile


AWS Amplify

SDKs


JavaScript


Python


PHP


.NET


Ruby


Java


Go

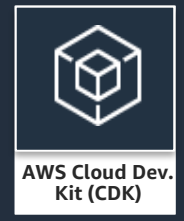

Node.js


C++

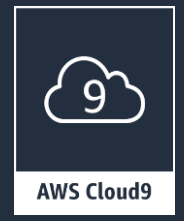
Comprehensive set of enterprise-grade tools

CI/CD Tools

Infrastructure as Code

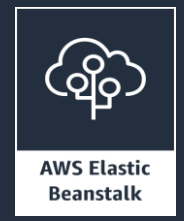


IDE



Monitoring & Tracing

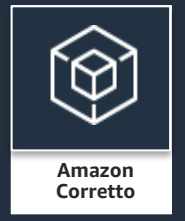
Web Apps



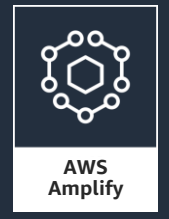
IDE and DevOps Toolkits

CLI and Scripting Tools

Languages

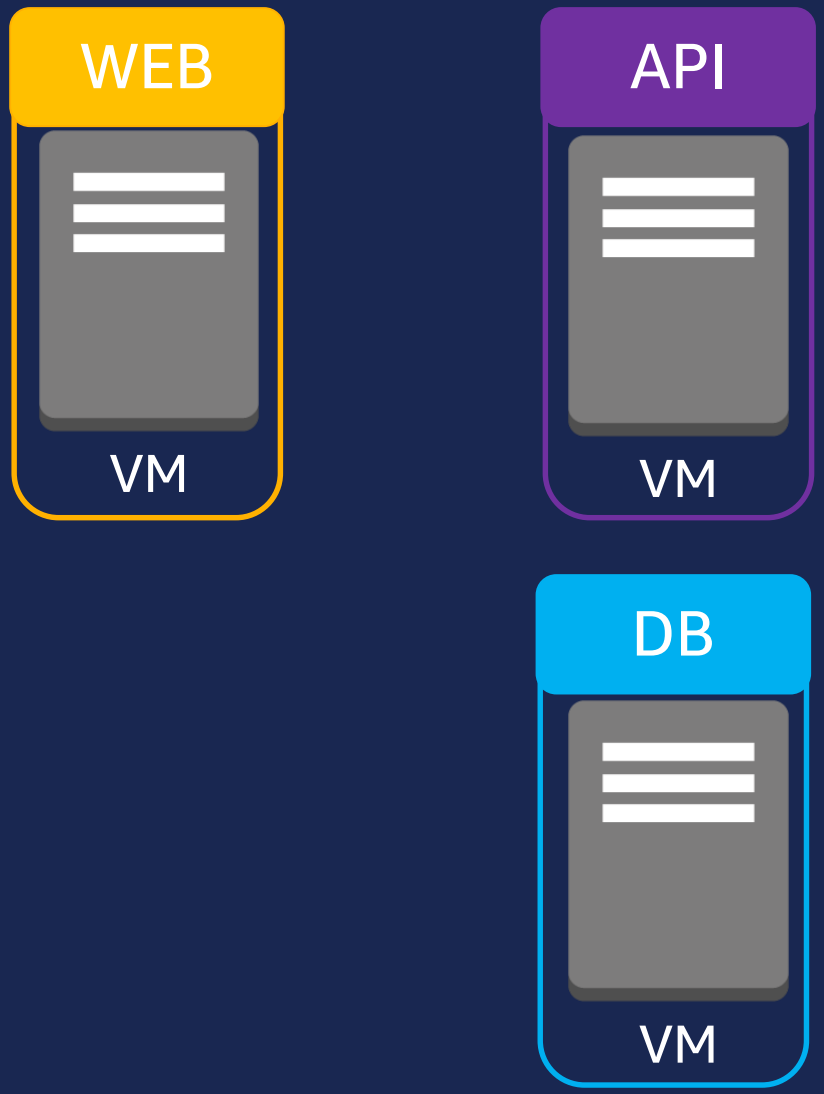


Mobile

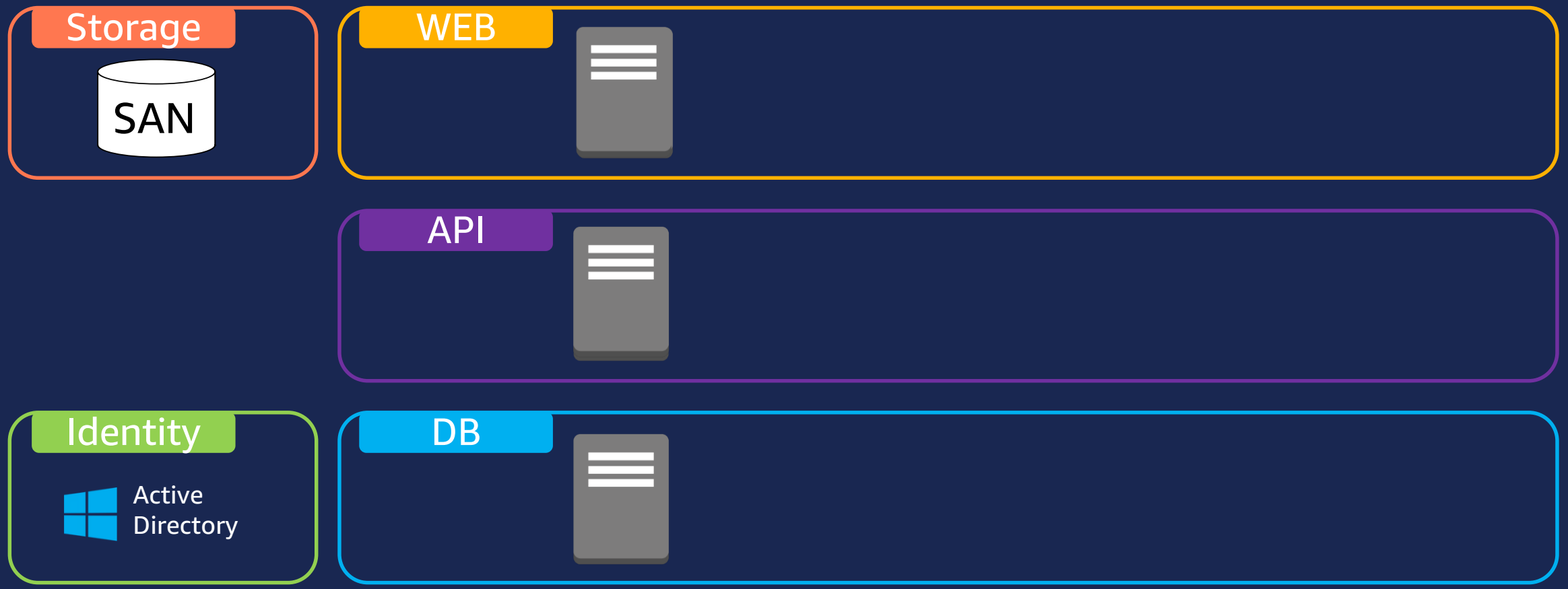


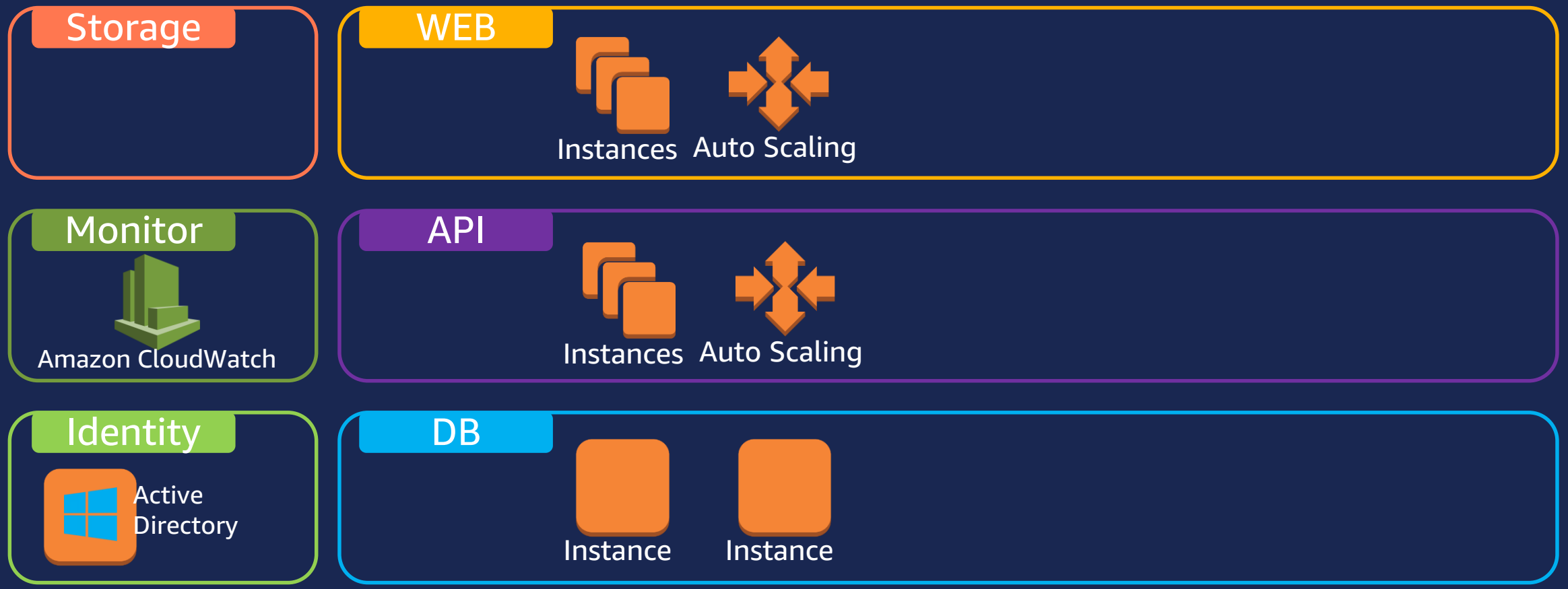
SDKs

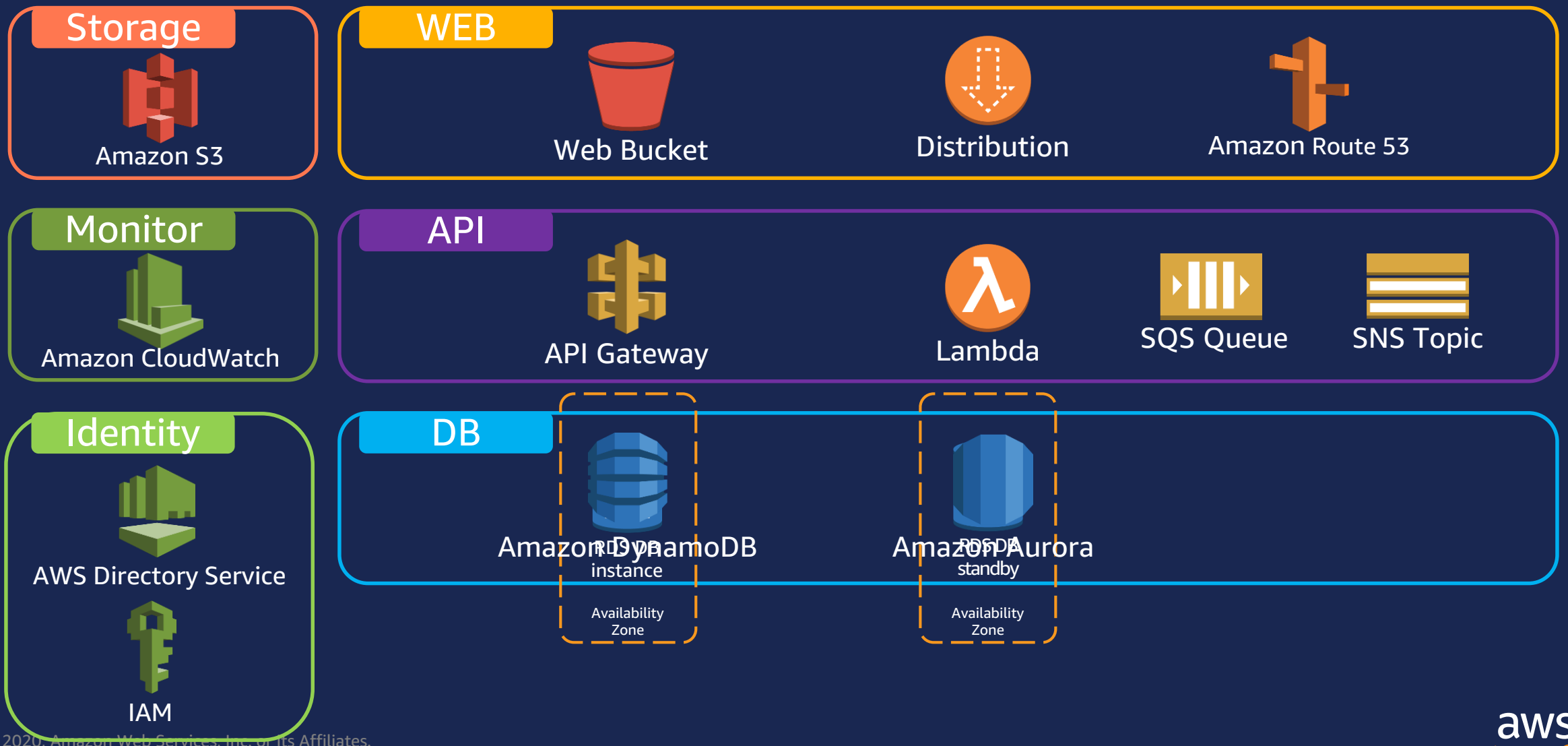
Application Arch based modernization / Progression in the Cloud…



Bank Web Property







AWS Compute Services for .NET

.NET Framework


Amazon EC2


AWS Elastic Beanstalk


Amazon Lightsail


Amazon Elastic Container Service


Amazon Elastic Container Service for Kubernetes

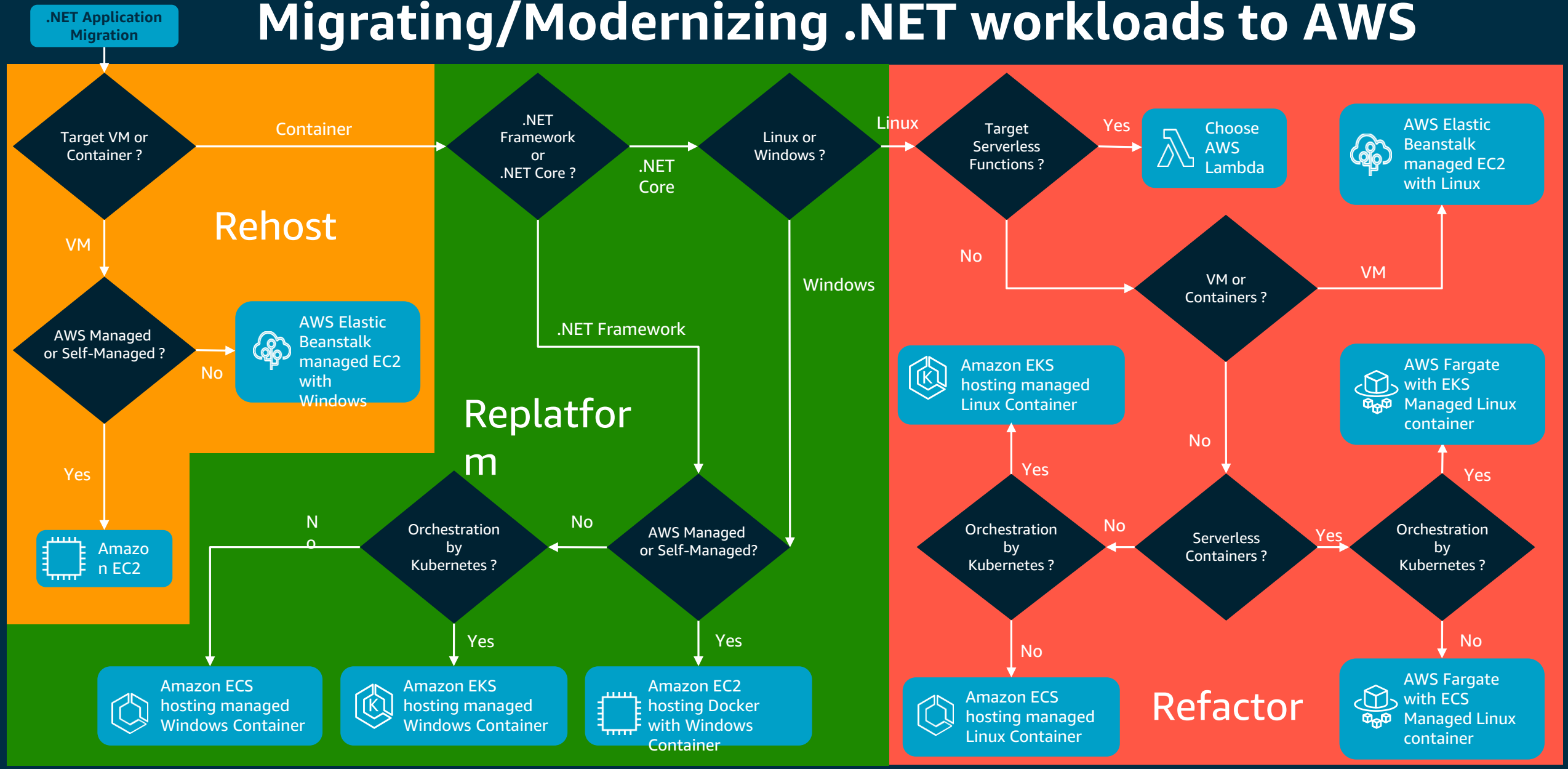

AWS Fargate


AWS Lambda

.NET 5/.NET Core

aws

Migrating/Modernizing .NET workloads to AWS



Application Modernization Tools

APP2CONTAINER

CONTAINERIZE AND MIGRATE
EXISTING APPLICATIONS

OPTIMUS

NEW

PREVIEW

PORTFOLIO ASSESSMENT AND GUIDED
MODERNIZATION DECISIONS

PORTING ASSISTANT FOR .NET

INSIGHT AND ASSISTANCE FOR PORTING
FROM .NET FRAMEWORK TO .NET CORE

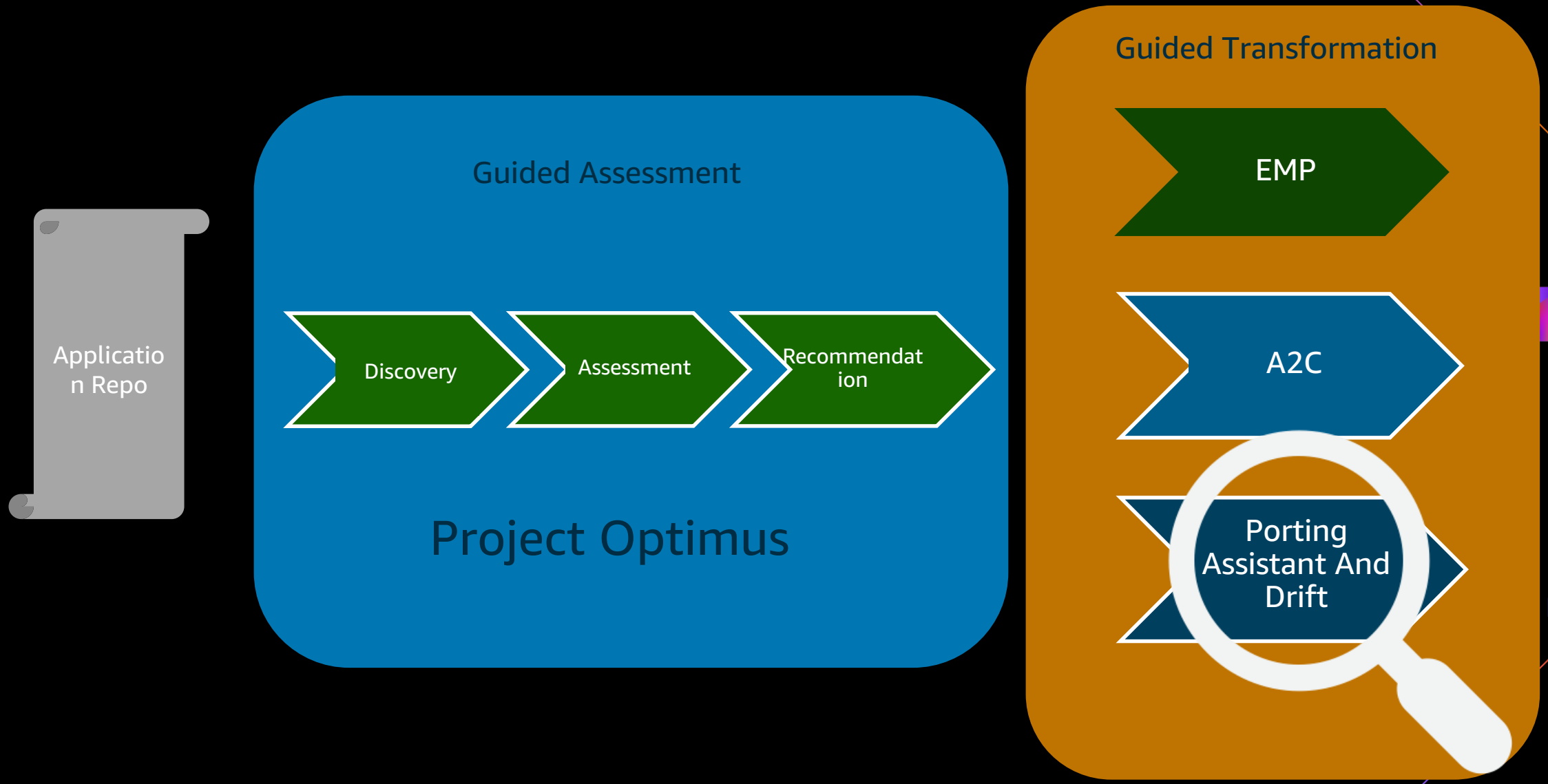
DRIFT

NEW

PREVIEW

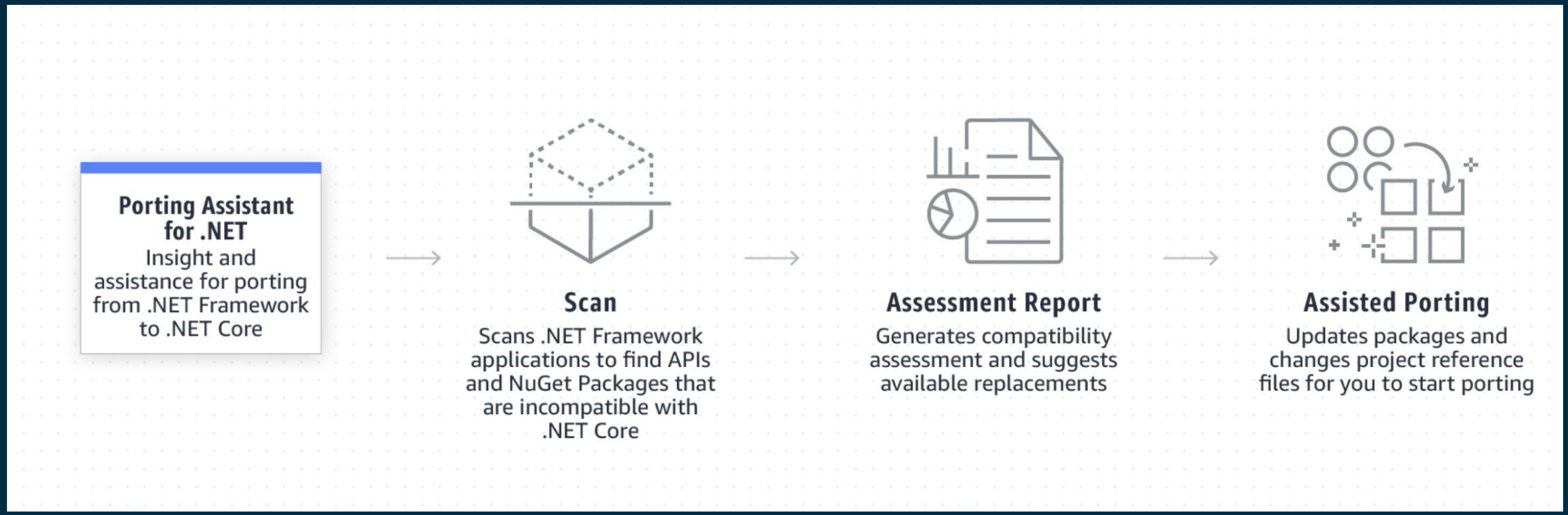
DECOMPOSE MONOLITHIC APPS TO MICROSERVICES

Refactor – Porting Assistant for .NET



Porting Assistant for .NET

Assessment and Porting .NET Framework app to .NET Core on Linux



AWS Porting Assistant for .Net : Port .NET framework to .NET Core

Porting Assistant for .Net is an analysis tool that scans .NET Framework applications and generates a .NET Core compatibility assessment, helping our customers port their applications to Linux faster

Benefits

- Reduces the manual effort involved in modernizing applications to Linux
- Makes it easy to prioritize applications for porting based on the level of effort required and compatibility assessment
- Accelerates application modernization

Porting Assistant for .NET is Open Source

Porting Assistant for .NET: How to start?

- Desktop Tool installed on developer Windows machine

- Prerequisites:
  - ❑ .NET Core 3.1
  - ❑ .NET Framework

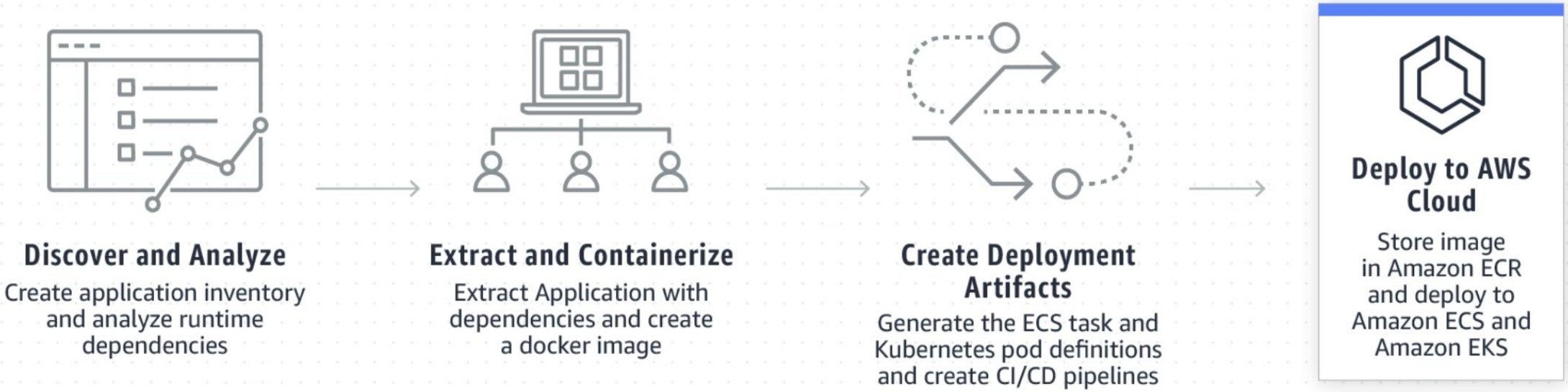
- Input : SLN file and source code of .NET Framework App

Porting Assistant for .NET: User Workflow

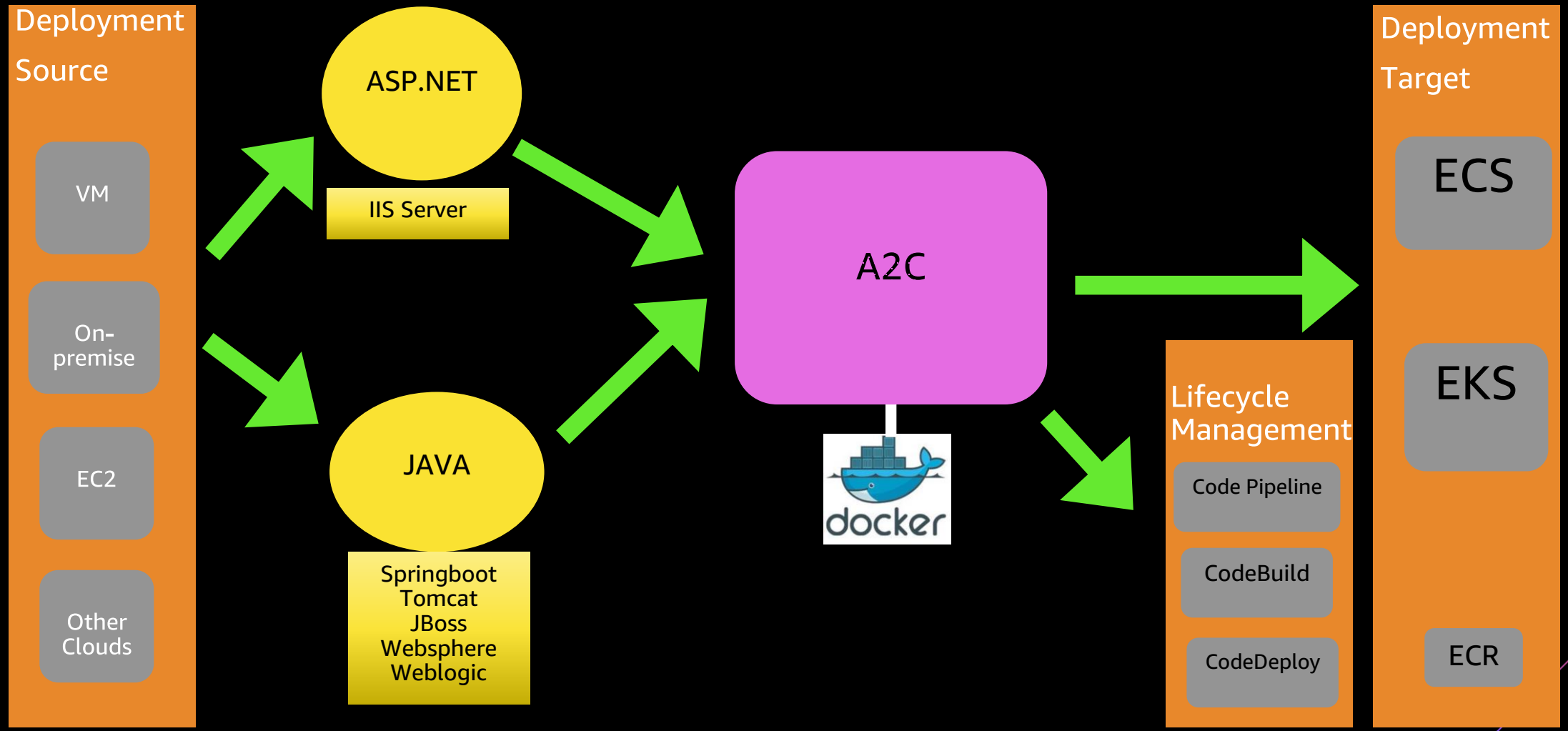
- Install prerequisites in developer machine with source code
- Download Porting Assistant Tool
- Run Porting Assistant Tool
- Key Functions :
  - Compatibility assessment of .NET framework app for .NET Core
  - Automated porting to .NET Core app
  - Deployment of .NET Core to Linux environment

AWS App2Container: How it works?

Transforming Applications running in VM to Containers and deploy to AWS



App2Container – Containerize and Migrate Existing Apps To AWS Cloud



App2Container CLI: User Workflow

1. Initialize – Configure App2Container tool
2. Inventory – Identify candidate applications running in VM
3. Analyze – Perform application dependency analysis for select application
4. Containerize – Create docker container for select analyzed application
5. Generate-Deployment – Generate AWS deployment artifacts on ECR and ECS task definition
6. Repeat steps-3-5 for each application
7. Split mode to accommodate unavailability of Docker tools on Windows Server 2008/2012
   - Steps 1-3 on Windows Server 2008/2012 Application Server
   - Steps 1,4-5 on Windows Server 2016/2019 – Worker Server for App2Container Tool

Guess… too many things in 45 mins..

Lastly….Few good resources to further read/explore…

Well Architected Framework

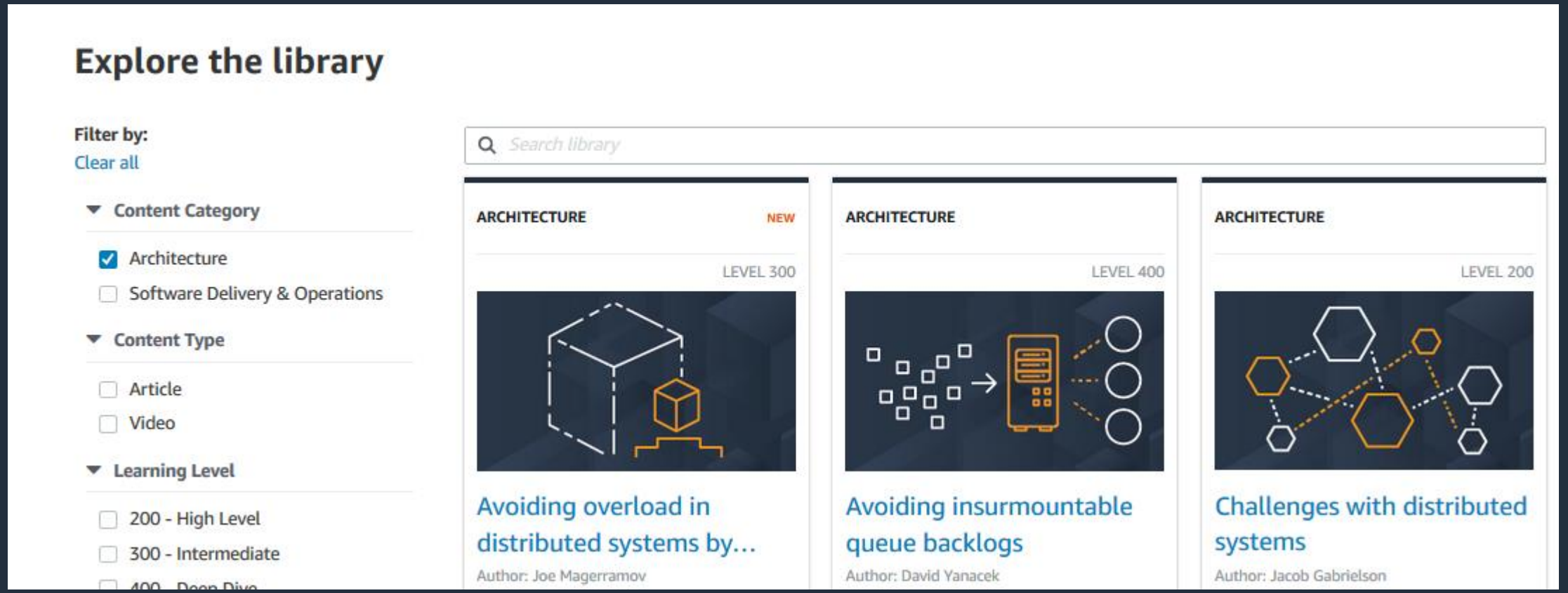
AWS Well-Architected

Learn, measure, and build using architectural best practices

- Operational Excellence
- Security
- Reliability
- Performance Efficiency
- Cost Optimization

The Amazon Builders' Library

How Amazon builds and operates software



12 Factor Apps (e.g. AWS Serverless lens)

The Twelve Factors

As you'll see, many of these factors are not only directly applicable to serverless applications, but in fact a default mechanism or capability of the AWS serverless platform. Other factors don't fit, and I talk about how these factors may not apply at all in a serverless approach.

I. Codebase
II. Dependencies
III. Config
IV. Backing services
V. Build, release, run
VI. Processes
VII. Port binding
VIII. Concurrency
IX. Disposability
X. Dev/prod parity
XI. Logs
XII. Admin processes

II. Dependencies

Explicitly declare and isolate dependencies

III. Config

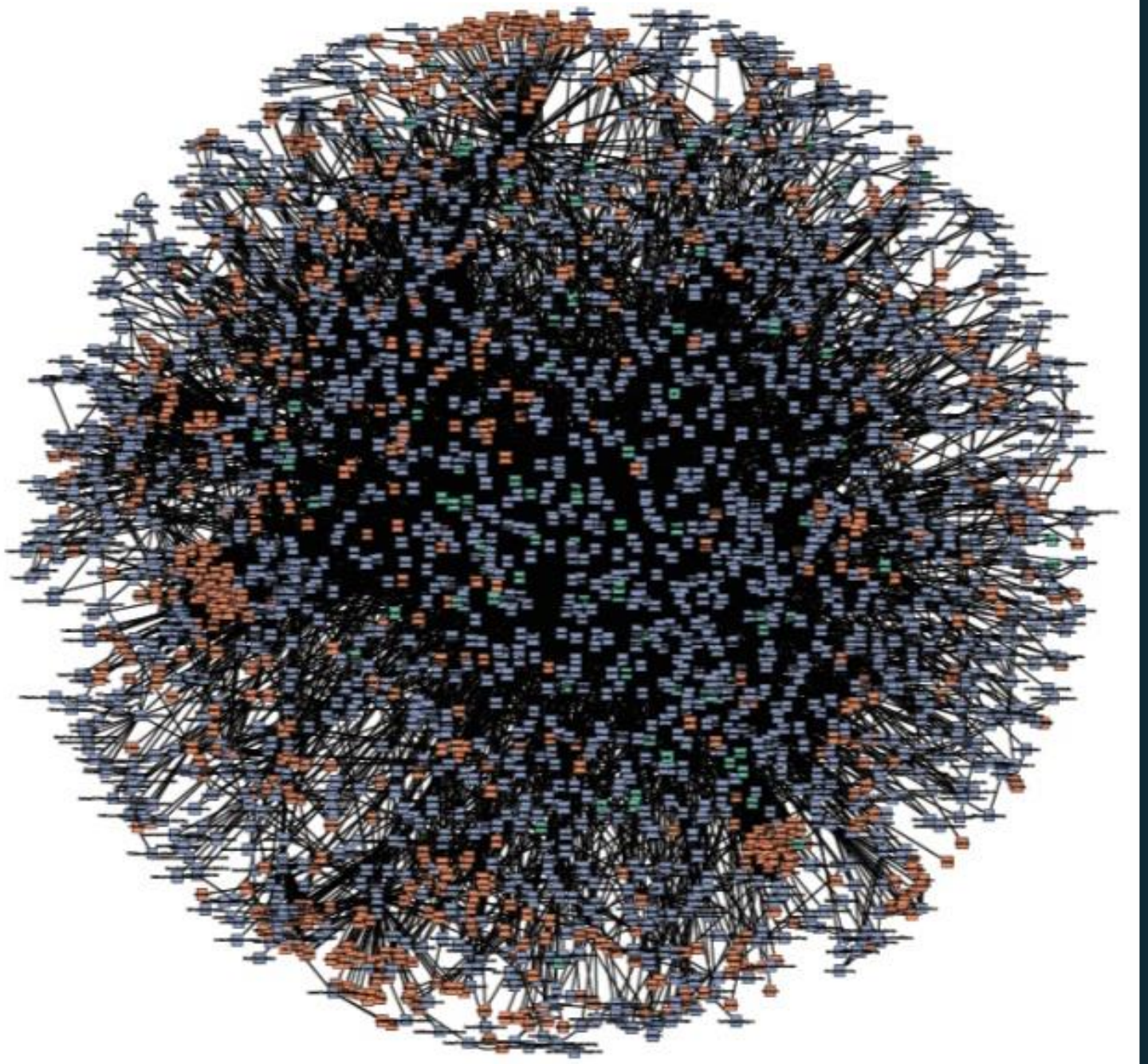
Store config in the environment

IX. Disposability

Maximize robustness with fast startup and graceful shutdown

XI. Logs

Treat logs as event streams



Everything gets a service interface

Primitives

“Microservices”

Polyglot (Language, Services, Tools)

Over the network vs in-memory

aws

Decouple state from code using messaging

IOT	Messaging	Managed	Streaming	Events
 AWS IoT Core	 Amazon SNS	 Amazon MQ	 Amazon Kinesis	 Amazon CloudWatch
	 Amazon SQS	 Amazon MSK		 Amazon EventBridge
		 Amazon ElastiCache (Redis)		

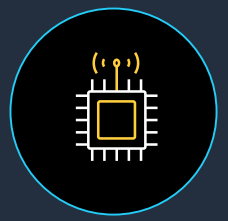
Purpose-built databases

Relational

Key-value

Wide Column

Document



In-memory

Graph

Time-series

Ledger

RDS

Aurora

Community

Commercial

Microsoft SQL Server

DynamoDB

Managed Cassandra Service

with Cassandra Compatibility

DocumentDB

with MongDB Compatibility

ElastiCache

Neptune

Timestream

QLDB

Useful links

AWS .NET homepage
https://aws.amazon.com/net

Open source .NET tools homepage
https://github.com/aws/dotnet

Developing and Deploying .NET Applications on AWS Whitepaper
https://bit.ly/2QCznmJ

.NET Core 3.0 on Lambda with AWS Lambda's Custom Runtime
https://amzn.to/35m5Rpp

Step Functions
https://amzn.to/2O8iChC

AWS Lambda Layers with .NET Core
https://amzn.to/2rhbx5a

.NET on AWS Labs (Immersion Day)
https://net-immersionday.workshop.aws

aws

Thank You