

AWS Cloud for .NET Developers

Amit Jha Developer Advocate

@amitkjha_rjn
@dotnetonAWS

© 2020, Amazon Web Services, Inc. or its Affiliates.



Agenda

AWS Primer AWS Tools for .NET Developers - Demo Different compute options for .NET Workloads on AWS - Demo Modernizing .NET Apps - Paths Supporting CI-CD, Tracing etc. Resources



What are your business goals?







Grow new revenue streams

Improve operational and financial efficiency

Lower business risk





Business drivers for embracing the cloud





AWS Global Infrastructure 24 geographical regions, 1 local region, 77 availability zones, 200+ POPs

Region & Number of Availability Zones (AZs)

Europe

Frankfurt (3), Paris (3),

London (3), Milan (3)

Middle East

Bahrain (3)

Ireland (3), Stockholm (3),

GovCloud (US) US-East (3), US-West (3)

US West Oregon (4) Northern California (3)

US East N. Virginia (6), Ohio (3)

Canada Central (3)

South America São Paulo (3)

Cape Town (3)

Africa

Asia Pacific Singapore (3), Sydney (3), Tokyo (4), Osaka-Local (1)* Seoul (4), Mumbai (3), Hong Kong (3)

China Beijing (2), Ningxia (3)

Announced Regions

Three Regions and 9 AZs in Indonesia, Japan, and Spain

* Available to select AWS customers who request access. Customers wishing to use the Asia Pacific (Osaka) Local Region should speak with their sales representative.



AWS Global Infrastructure

Build once, 245 Countries & Territories Served **AWS Cloud** 220 POP (Point of presence) run globally **97 Direct Connect Locations** 24 + 3 announced DR, Global Needs **AWS Region** 2...n AWS Availability Zone-77 HA, DR 2...* **AWS Data Center** Active Regions O Announced Regions x...z (e.g. 100,00**0**+) Performance Physical Server https://aws.amazon.com/about-aws/globalinfrastructure/regions az/ 1... n **EC2** Instances

Ensuring High Availability



The AWS Cloud infrastructure:

- A Region is a physical location in the world where we have multiple Availability Zones
- Availability Zones (AZ) consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities. Applications and Data are replicated in real time and consistent in the different AZs
- AWS Local Zones place compute, storage, database, and other select AWS services closer to end-users.

Distance

ensures high

availability

Low-latency

ensures real data

replication

AWS Services

More services and more functionality within those services

ANALYTICS	
ANALYTICS DATA EXCHANGE DATA LAKE DATA PIPELINES DATA WAREHOUSE ELASTICSEARCH	STREAMING ETL HADOOP/SPARK INTERACTIVE SQL QUERIES VISUALIZATIONS

AR + VR

AR/VR EXPERIENCES

AWS COST MANAGEMENT

ANALYZE AWS COSTS COST & USAGE BUDGETS COST & USAGE REPORTS RESERVED INSTANCES REPOR

좌 APPLICATION INTEGRATION

EMAIL	SEARCH
MESSAGE BROKER	TRANSCODING
QUEUEING & NOTIFICATIONS	WORKFLOW

BUSINESS APPLICATIONS

EMAIL & CALENDARING	UNIFIED COMMUNICATIOI
ONLINE MEETINGS	VOICE-ENABLED
SHARING &	WORKPLACE

DE BLOCKCHAIN

BLOCKCHAIN TEMPLATES LEDGER DATABASE MANAGED BLOCKCHAIN

CUSTOMER ENGAGEMENT

CONTACT CENTER EMAIL TARGETING USER ENGAGEMENT ACROSS CHA

	1	
	🔲 СОМРИТЕ	
	AUTO SCALING	CONTAI
	BATCH JOBS	MANAG
	EVENT-DRIVEN	STOPE
	COMPUTING	DOCKER
	INSTANCE TYPES	
_	MANAGED VIRTUAL	
	MANAGED	
	REPOSITORY FOR SERVERLESS APPS	
	RUN & MANAGE	
	WEB APPS	
	SERVERLESS	

NER SERVICE

VIRTUAL SERVERS

REALTIONAL DATABASES PURPOSE-BUILT DATABASE HIGH-PERFORMANCE RELATIONAL DATABASE DOCUMENT DATABASE DATABASE DATABASE MANAGED MARIADB IN-MEMORY CACHING MANAGED MYSQL KEY-VALUE STORE DATABASE MANAGED ORACLE LEDGER DATABASE MANAGED POSTGRESQL TIME SERIES DATABASE MANAGED SQL SERVER TATABASE

DEVELOPER TOOLS

ANALYZE & DEBUG APPLICATION LIFECYCLE MANAGEMENT AUTHORING BUILD & TEST CONTAINERS DEVOPS RESOURCE MANAGEMENT ONE-CLICK APP DEVELOPMENT PATCHING PIPELINE ORCHESTRATION RESOURCE TEMPLATES TRIGGERS

END USER COMPUTING									
APP STREAMING DESKTOP COMPUTING	MOBILE ACCESS STORAGE & COLLABORATION								
A HYBRID AR	CHITECTURE								
AWS SERVICES INTEGRATED									



တို့တ္ GAME TECH

CROSS-PLATFORM 3D GAME ENGINE GAME SERVER HOSTING

A INFRASTRUCTURE

AVAILABILITY ZONES CUSTOM HARDWARE DATA CENTER INFRASTRUCTURE GLOBAL NETWORK BACKBONE POINTS OF PRESENCE POWER INFRASTRUCTURE REGIONS

INTERNET OF THINGS (IOT)

RULES ENGINE DEVICE ANALYTICS DEVICE GATEWAY DEVICE SDK DEVICE SHADOWS EVENT DETECTION & RESPONSE LOCAL COMPUTE LOCAL DATA COLLECTION MANAGEMENT & SECURITY MICROCONTROLLER OPERATING SYSTEM REGISTRY VISUAL APPLICATIONS DEVELOPMENT

∰ MACHINE LE	ARNING
MACHINE LE MIL FRAMEWORKS DEEP LEARNING AMIS & CONTAINERS HARDWARE ACCELERATION ML AT THE EDGE TENSORFLOW, PYTORCH, MXNET ALSERVICES CHATBOTS ENTITY EXTRACTION FACE ANALYTICS FACE SEARCH FORECASTING IMAGE LABELING NATURAL LANGUAGE PROCESSING PERSONALIZATION & RECOMMENDATION SENTIMENT ANALYSIS SPEECH TRANSCRIPTION TEXT & DATA EXTRACT	ARNING AUTOMATIC MOD TUNING DATA LABELING HOSTED NOTEBO ML MARKETPLACE MODEL HOSTING MODEL HOSTING PRE-BUILT ALGORITHMS TOPIC MODELING DEEP LEARNING MODELS REINFORCEMENT LEARNING SPOT INSTANCESS BATCH PREDICTIONS ION
TRANSLATION	
TRANSLATION	
VIDEO & IMAGE ANALY	SIS
CONTENT MODERATION	N

MANAGEMENT & GOVERNANCE

ACTIVITY & API MONITORING USAGE TRACKING PROVISIONING CHATBOT RESOURCE CONFIGURATION TEMPLATES TRACKING SECURITY GOVERNANCE RECOMMENDATIONS INVENTORY SERVER TRACKING MANAGEMENT LICENSE MANAGER SERVICE CATALOG MANAGE POLICIES SYSTEMS MANAGER MANAGE RESOURCES

MARKETPLACE

ANALYTICS MACHINE LEARNING DATA PRODUCTS NETWORKING DATABASES OPERATING SYSTEMS DEVOPS SECURITY IOT STORAGE

DJ MEDIA SERVI

LIVE VIDEO VIDEO VIDEO VIDEO TRANSPORT PERSO MODIA STORAGE MONE TRANSCODING VIDEO ORIGINATION VIDEO ORIGINATION VIDEO ORIGINATION ONDE

MIGRATION & TRANSFI

APPLICATION MIGRATION DATABASE MIGRATION EXABYTE-SCALE MIGRATION ONLINE DATA TRANSFER SCHEMA CONVERSION SERVER MIGRATION TRANSFER FOR SFTP

MOBILE

API GATEWAY MOBILE APP TESTING DEVELOPMENT SINGLE INTEGRATED FRAMEWORK CONSOLE IDENTITY SYNC MOBILE ANALYTICS TARGETED PUSH NOTIFICATIONS

CONTENT DELIVER

APPLICATION DELIVERY DEDICATED NETWORK CONNECTION DOMAIN NAME SYSTEM LOAD BALANCING MONITOR APIS MONITOR MICROSERVICES NETWORK TOPOLOGY NETWORKING HUB PRIVATE CONNECTION TO APPS SCALE VPC & ACCOUNT CONNECTIONS SERVICE DISCOVERY VIRTUAL PRIVATE CLOUD

CLOUD ROBOTICS

SATELLITE OPERATIONS

SECURITY, IDENTITY, & COMPLIANCE

ACCESS CONTROL ASSESSMENT & REPORTING CONFIGURATION COMPLIANCE DATA PROTECTION DDOS PROTECTION IDENTITY MANAGEMENT KEY MANAGEMENT & STORAGE MONITORING & LOGGING RESOURCE MANAGEMENT THREAT DETECTION WEB APPLICATION FIREWALL

STORAGE

ARCHIVE STORAGE BACKUP & RESTORE BLOCK STORAGE DATA TRANSFER EDGE PROCESSING & COMPUTING FILE STORAGE HIGH-PERFORMANCE FILE SYSTEM HYBRID CLOUD STORAGE OBJECT STORAGE WINDOWS FILE SYSTEM

CUSTOMER ENABLEMENT

ACCOUNT MANAGEMENT DASHBOARD PERSONALIZATION ENTERPRISE SUPPORT EXPERTS MARKETPLACE OPTIMIZATION GUIDANCE PARTNER ECOSYSTEMS PROFESSIONAL SERVICES SECURITY & BILLING REPORTS SOLUTIONS MANAGEMENT TRAINING & CERTIFICATION

Shared responsibility model





AWS security, identity, and compliance solutions





The AWS ML stack

Broadest and most complete set of machine learning capabilities





Data Analytics







An Expansive Ecosystem

Products integrated with AWS platform and easy to test



There are tens of thousands of AWS Partners across the globe.

More than 90% of Fortune 100 companies and the majority of Fortune 500 companies use AWS Partner solutions and services.

Quick Starts are built by Amazon Web Services (AWS) solutions architects and partners to help you deploy popular technologies on AWS, based on AWS best practices for security and high availability. There are more than 200 Quick Starts published.



AWS Marketplace is available in 24 AWS Regions.

AWS Marketplace connects customers with more than 1,600 independent software vendors to meet their business needs.

More than 300,000 active customers are using software from AWS Marketplace.

AWS Marketplace simplifies the procurement, entitlement, and provisioning of software across 50 categories, totaling more than 8,000 transactable listings.

There are more than 1.5 million current software subscriptions from AWS Marketplace.

AWS customers use over 850 million hours a month of Amazon EC2 for AWS Marketplace products.

Overview of .NET on AWS

© 2020, Amazon Web Services, Inc. or its Affiliates.





AWS Tools for .NET Developers

© 2020, Amazon Web Services, Inc. or its Affiliates.



Development Tools







Demo



IDE Integration

© 2020, Amazon Web Services, Inc. or its Affiliates.



AWS Toolkit for Visual Studio



AWS Explorer tool window and rich views

- Presents tree view of AWS services and resources of interest to developers
- Manage S3 Buckets
- View, create, edit, and delete Amazon DynamoDB tables
- Launch and manage (including simple RDP into) EC2 compute instances
- ...and much more!



Wizards to easily deploy applications

- AWS Elastic Beanstalk
- Amazon Elastic Container Service
- AWS Lambda
- AWS CloudFormation



"Getting started" project templates ("blueprints") for serverless functions and applications



Code-editing Intellisense for AWS CloudFormation json templates

https://aws.amazon.com/visualstudio/

AWS Toolkit for Visual Studio

AWS Toolkit for Visual Studio

AWS Toolkit for Visual Studio



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



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

00	•

Deployment from the IDE

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



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

00	••		
	•		
	•	 (_	7)
			7)

Deployment from the IDE

	Ì
	/
\sim	
У 	

Access CloudWatch From the IDE

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



Programmable SDK

© 2020, Amazon Web Services, Inc. or its Affiliates.



AWS SDK for .NET





Central home for all SDK and extension libraries https://github.com/aws/dotnet



Programmatic access to all AWS services

- Updates almost daily (in sync with service updates and launches)
- Simple programming model



Various extension libraries cover 'higher level' functionality



Support .NET cross platforms

- Supports .NET Framework, .NET Core, and Xamarin development
- Open source



Distributed via NuGet





Consistent SDK Coding Pattern

```
using servicename;
using servicename.Mode1;
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
}
```



Real-world example—list all objects in an S3 bucket

```
using Amazon.S3;
 4
     using Amazon.S3.Model;
    pnamespace ConsoleApp
         class Program
              static async Task ListS3BucketObjects()
11
12
                 using (var s3Client = new AmazonS3Client(Amazon.RegionEndpoint.USWest2))
14
                      string nextPageMarker = null;
15
                      do
17
                         var listObjectsResponse = await s3Client.ListObjectsAsync(new ListObjectsRequest
19
                              BucketName = "my-bucket-name",
                              Marker = nextPageMarker
21
                          });
22
23
24
                          nextPageMarker = listObjectsResponse.NextMarker;
25
                          foreach (var s30bject in list0bjectsResponse.S30bjects)
27
                              Console.WriteLine($"{s30bject.Key}, last modified {s30bject.LastModified.ToString("g")}");
29
                       while (!string.IsNullOrEmpty(nextPageMarker));
```





Demo...



AWS offers lots of options



Compute: EC2, Lambda, ECS, EKS, Lightsail, Fargate



Storage: S3, Elastic File Service



Network: Classic Load Balancer, Application Load Balancer, Network Load Balancer, Security Groups, Virtual Private Cloud



Database: RDS, DynamoDB, Aurora, Neptune



Management: Auto Scaling, CloudFormation, Systems Manager, CloudWatch

Typical Application Journey











Amazon EC2

- Virtual Machines (Linux/Windows)
- Secure
- Resizable



AWS Lambda

- Execute code without provisioning or managing servers
- Synchronous (request/response) or Asynchronous execution models



Containers

 Broad range of fully managed, do-it-yourself and serverless container options



AWS Compute Services for .NET

.NET Framework



AWS Toolkit options for .NET 5 in Lambda

🚺 New AWS Lambda C# Proiect

X

aws Select Blueprint Choose the contents of the C# project for your AWS Lambda function.

Blueprints are Lambda projects that contain getting started code for your functions and a test project. Choose a blueprint that best aligns with your desired scenario and customize as needed.

Filter:

function from scratch.

Custom

Detect Image Labels

Use Amazon Rekognition service to tag image files in S3 with detected labels.

S3, Rekognition

Simple Application Load... A skeleton Lambda function to get started responding to Application Load Balancer





Finish

Back



From the command line

If you don't have AWS templates for .NET installed, install them:

> dotnet new -i Amazon.Lambda.Templates

Then, create a new .NET Lambda container image project:

> dotnet new lambda.image.EmptyFunction --output <project-name> --region <aws-region>

Install the Amazon.Lambda.Tools tool if you don't already have them:

> dotnet tool install -g Amazon.Lambda.Tools

Deploy using the tools:

- > dotnet lambda deploy-function
- For serverless application templates, you can deploy with deploy-serverless
- Equivalent templates and commands let you do the same with custom runtimes



EC2 Linux AMI with .NET 5

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Q .net Core 5.0			×
		Search by Sy	vstems Manager parameter
Quick Start (1)		I< < 1	to 1 of 1 AMIs >>
My AMIs (0)	Ũ	Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE	Select
AWS Marketplace (2)	Amazon Linux Free tier eligible	.NET Core 5.0, Mono 6.12, PowerShell 7.1, and MATE DE pre-installed to run your	64-bit (x86)
Community AMIs (5)		.NET applications on Amazon Linux 2 with Long Term Support (LTS).	



Cancel and Exit

Compute Resources: Where to Run Code



EC2 Instances: Windows/Linux (.NET Framework, .NET Core)

- Choose instance type/size, memory, disk, vCPUs
- You manage and patch



Containers: Windows, Linux (.NET Framework, .NET Core)

- Easy to package, deploy, reuse and scale
- Focus on development, not infrastructure



Lambda (.NET Core)

- Serverless platform for short-running tasks (< 15 min)
- Pay only for the time the code is executing
- Choose from C#, Node.js, Python, Java, Go, Ruby, or bring your own runtime!



Update to .NET Core When Possible



.NET Framework Legacy Monolithic Windows-only



Porting Assistant for .NET



.NET Core (.NET 5)

Active development

Lightweight & flexible

Cross-platform



To port, or not to port, that is the question.

Consider porting an app to .NET Core to:

- Take advantage of new & future innovations (ex: gRPC, WebAssembly)
- Reduce licensing spend (no Windows tax)
- Deploy to Lambda, Fargate, IoT

Candidates for porting:

- ASP.NET web apps:
 - MVC
 - Web API
- Console apps
- New features being developed has a future
- No Windows-specific dependencies

Not so great candidates*:

- Legacy ASP.NET web apps:
 - Web Forms
 - WCF server apps*
- Desktop apps
- Not being actively developed in maintenance mode
- Has other Windows dependencies



Assessment – Packages and API calls

Assessment overview

The level of compatibility will affect the effort required to port your solution to .NET Core.



View dependency graph

Project references graph Info

The following projects are most referenced by other projects. Use this graph to determine which project to port first. Select a node to see its project dependencies.

View details



aws

© 2020, A

Demo....



App2Container (A2C)

Package .NET Framework applications into containers

No code changes

ASP.NET Web applications

- .NET 3.5+
- IIS 7.5+

Windows containers





Containerize & deploy .NET Framework apps





App2Container Features

Application Inventory and analysis:

- Identifies all the ASP.NET apps running on IIS
- Identifies dependencies
- Finds database connection strings and prompts to replace

Easy automated deployment and CI/CD

- Creates the container image, and pushes it to Amazon ECR
- Generates ECS Task defs or Kubernetes deployment YAML
- Creates CI/CD pipeline for automated deployments

Infrastructure automation

Generates CloudFormation templates that provision everything!

Also works with Java apps on Linux (Tomcat, Spring Boot, JBoss, WebLogic...)

Other – .NET App Modernizations



Amazon Redis ElastiCache based Session State





Using Amazon ElastiCache with Microsoft RedisSessionStateProvider

- Launch a Redis cluster in Amazon ElastiCache. Make note of endpoint
 - address for the cluster. (in cluster mode enabled, the endpoint is called "configuration endpoint" and in the cluster mode disabled, the endpoint is called "primary endpoint")
- Open ASP.NET project in Microsoft Visual Studio. Use Nuget Package Manager to install "RedisSessionStateProvider" package.



Microsoft.Web.RedisSessionStateProvider S by Microsoft, **4.11M** di v4.0.1 Custom session state provider for redis cache.



Using Amazon ElastiCache with Microsoft RedisSessionStateProvider

• In the ASP.NET project, Update "RedisSessionStateProvider"

configuration in the "web.config" file. Update the "host" with the cluster endpoint address.

<sessionState mode="Custom" customProvider="MySessionStateStore">

<providers>

<add name="MySessionStateStore" type="Microsoft.Web.Redis.RedisSessionStateProvider"</pre>

```
host="<Endpoint_Address>" accessKey="" ssl="false" />
```

</providers>

</sessionState>

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.*

Identify performance bottlenecks: Workflow View





Identify performance bottlenecks: Request Details

Traces > De	etails																	
Timeline	Raw data																	
Method POST	Response 201	Duration Age ID 2.1 sec 36.3 min (2018-03-16 20:13:19 UTC) 1-5aac255f-f211cd8deb4c610083aed926																
Name			R	les.	Duration	Status	0.0ms	200ms	400ms	600ms I	800ms I	1.0s	1.2s	1.4s	1.6s	1.8s	2.0s	2.2s
▼ mysignupf	rontend.2mbmte	pd39.us-west-2	2.elastict	beanst	alk.com AWS	S::ElasticBe	anstalk::En	vironment										
mysignupfr	ontend.2mbmtep	d39.us-west-2.e	elastict	201	2.1 sec	\checkmark												POST /remoteSignup
mysignu	papi.2mbmtepd3	9.us-west-2.elas	sticbea	201	2.1 sec	\checkmark												Remote: POST /signup
mysignupa	pi.2mbmtepd39	us-west-2.elas	ticbeans	stalk.co	om AWS::Elasti	icBeanstall	c:Environm	ent										
mysignupa	pi.2mbmtepd39.u	s-west-2.elastic	beans	201	2.1 sec													POST /signup
Dynamo	DB			200	2.1 sec	\checkmark												PutItem: awseb-e-96vjw2bqv9-stack-StartupSignupsTable-U2AMY
SNS				200	31.0 ms	\checkmark	i											Publish
DynamoDE	AWS::DynamoDB:	Table (Client Resp	onse)															
mysignupa	pi.2mbmtepd39.u	s-west-2.elastic	beans	200	2.1 sec													Putitem: awseb-e-96vjw2bqv9-stack-StartupSignupsTable-U2AMY
▼ SNS AWS::S	SNS (Client Respons	e)																
mysignupa	pi.2mbmtepd39.u	us-west-2.elastic	beans	200	31.0 ms												-	Publish

Application Insights for .NET and SQL Server

© 2020, Amazon Web Services, Inc. or its Affiliates.



How it works



Application discovery and configuration Data preprocessin g Intelligent problem detection Alert and action

Application Components

- Compute
 - Amazon EC2
 - AWS Lambda
- Database
 - Amazon RDS
 - Amazon DynamoDB
- Groups
 - Elastic Load Balancing
 - Amazon EC2 Auto Scaling group
 - Custom Resources
- Other services
 - Amazon Simple Queue Service
 - Amazon S3 bucket metrics

Standa	atorie and grouped resources in the application	on.			1
Q					
	Component name	Description ∇	Туре 🗢	Resources	Monitoring
0	Servi-lista-QGNX2VQO21IH	Application Load Balancer group	Auto-grouped	View	⊘ Enabled
0	Servi-payfo-1DGD3UJ769Z8P	Application Load Balancer group	Auto-grouped	View	⊘Enabled
0	Servi-petsi-O1QM5X2AXESZ	Application Load Balancer group	Auto-grouped	View	⊘ Enabled
0	Servi-searc-1XIXVTU8VYD3U	Application Load Balancer group	Auto-grouped	View	⊘Enabled
0	Services- AWS679f53fac002430cb0da5b7982 bd22872D164C-123W501Q26O9	Lambda function	Standalone	View	▲ Not enabled
0	Services- CustomCDKBucketDeployment8693B B64968944B6-10I5ODK2JMEGF	Lambda function	Standalone	View	▲ Not enabled
0	Services-ddbpetadoption7B7CFEC9- MKAA6MOQ4BGT	DynamoDB table	Standalone	View	⊘ Enabled
0	Services- ddbseederpetadoptionhandler8986B 6EA-62XR2531TZXC	Lambda function	Standalone	View	▲ Not enabled
0	Services- sqlseedersqlseederlambda83C15CB8 -17CJBYNINKEZM	Lambda function	Standalone	View	▲ Not enabled
0	Services- sqlseedersqlserverseederproviderfra mework-XEYZOAPUEELU	Lambda function	Standalone	View	<u>∧</u> Not enabled



Dashboards and insights

- Dashboards •
- Alerts ullet
- Insights •
 - Summary ullet
 - Start time and date \bullet
 - Severity ullet
 - Status \bullet
 - Possible root cause \bullet
 - **Related observations** \bullet

Recent alarms [•] View rece	ent alarms dashboard
ApplicationInsights/P 9	ApplicationInsights/P 9
OK Count	Count
1 47.5k ApproximateNumberOfM	346
2 01:20 04:20	5 ConsumedReadCapacit 01:20 04:20
4 ApproximateNumberOfMessages\	ConsumedReadCapacityUnits
3 ApplicationInsights/P	ApplicationInsights/P
- Count	2
- 382 194	1 - FailedSQLServerAgentJ
5 <u>ConsumedWriteCapacit</u> 01:20 04:20 ConsumedWriteCapacityUnits	0 01:20 04:20 FailedSQLServerAgentJobsCount
	ApplicationInsights/P ApplicationInsights/P 0K Count 52.2k 4 4 01:20 04:20 4 01:20 04:20 ApplicationInsights/P 0 Count 01:20 04:20 ApplicationInsights/P 0 Count 382 0 194 5 ConsumedWriteCapacit 0 01:20 04:20 04:20

Problems detected					
Severity	Problem summary	Source	Start-time 👻	Status	^
Medium	SQS: Queue Length	Services-sqspetadoption2E8B1217-VK0SN7FYM7M3	2020-06-09T05:46:41Z	In progress	



Dashboards and insights - example

Problem sur	nmary					
Severity	Problem summary	Source	Start-time	Status	Resource group	
Low	ALB: Backend 4XX errors	Servi-petsi-O1QM5X2AXESZ	2020-06-14T00:37:34Z	In progress	PetSite	
Insight ① Target instan logs or acces requests dire	ces sent 4XX responses to as logs, you can view them t actly to the instance and by	the Application Load Balancer. to determine the cause of the e bypassing the load balancer to s insight is useful This	If you configured your app rror. Otherwise, view these view the responses.	plication to ad e logs on your	d your application error instances. You can send	
			,			
Application Load Balancer group - app/Servi-petsi-O1QM5X2AXESZ						
app/Servi-p	etsi-O1QM5X2AXES2	C-HIIPCo ⊘				
No unit						
0.5 0	= 0.1 for 1 datapoints within 5 m	inutes				
	02:00 0	05:00 08:00				



Secure Development Environments in AWS



AWS CodePipeline – Continuous Delivery

Define and model complex application release workflows with multiple stages Automate builds, tests, and deployments with a visual interface Receive immediate feedback on failures to enable rapid iteration Integrate with existing build tools and test services

Integrates with third-party services:





The AWS Machine Learning Stack

Broadest and most complete set of ML capabilities

AI SERVICES



ML FRAMEWORKS & INFRASTRUCTURE

TensorFlow mxnet	GLUON K Ker Clearth DeepGraph	r as Deep Learning _{Library} AMIs & Containers	GPUs & CPUs	Elastic Inference	Inferentia	FPGA
						$\gamma \lambda \lambda c$



Amazon Translate – AWS SDK for .NET

var translateClient = new AmazonTranslateClient(Amazon.RegionEndpoint.USEast1);

```
var translateRequest = new TranslateTextRequest
```

```
Text = "Hello, how are you?",
SourceLanguageCode = "en", // English
TargetLanguageCode = "es" // Spanish
};
```

// Call TranslateText to convert the text
var translateResponse = await translateClient.TranslateTextAsync(translateRequest);

```
// Obtain the translated text in Spanish
string outputString = translateResponse?.TranslatedText;
```

Console.WriteLine(outputString);

[Console output] Hola, ¿cómo estás?

Services, Inc. or its affiliates. All rights reserved.



Demo...



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

Step Functions

https://amzn.to/2O8iChC

AWS Lambda Layers with .NET Core

https://amzn.to/2rhbx5a





Thank you!

Amit Jha Developer Advocate

@amitkjha_rjn
@dotnetonAWS

© 2020, Amazon Web Services, Inc. or its Affiliates.

