

AWS Webinar

Serverless Application Debugging and Delivery Best Practices

Steve Mayszak

19, September 2018

Steve Mayszak – SDM – AWS Cloud9

Software Development Manager @ AWS since 2014

- Presently: AWS Cloud9
- Previously: AWS CodeDeploy

Before Amazon: (1997 – 2014)

- Getty Images, Docusign, Nordstrom (and others)

Specialties'

- .net developer since it was first released
- Search: Lucene and proprietary projects.
- Video Transcoding
- Web Development

Hobbies:

- Music – playing guitar, listening, watching
- Movies – “Grew up” in a video store
- Cycling – island adventures are my favorite
- Traveling – Been to 5 continents & 17 countries
- Complaining about Seattle –it’s very dark....



Webinar Format

- 1) Not many slides.
- 2) Mostly live demo (from AWS.Console)
- 3) Demo: How to use AWS Developer tools for Lambda to:
 - 1) Create
 - 2) Author/Debug
 - 3) Release
 - 4) Post-Release/Operations/Troubleshooting
- 4) Come back to the slide deck at the end

Tool overview



SAM – Serverless Application Model

- 1) DSL over CloudFormation
- 2) Defines a Lambda Application
- 3) Organizes all AWS resources related to your App.
- 4) Tools for create / debug
- 5) Open source project:

<https://github.com/aws-labs/serverless-application-model>

SAM – Example:

`AWSTemplateFormatVersion: '2010-09-09'`

`Transform: 'AWS::Serverless-2016-10-31'`

`Description: An Amazon SNS trigger that sends CloudWatch alarm notifications to Slack.`

`Resources:`

`cloudwatchalarmtoslack:`

`Type: 'AWS::Serverless::Function'`

`Properties:`

`Handler: index.handler`

`Runtime: nodejs4.3`

`CodeUri: .`

`Description: >-`

`An Amazon SNS trigger that sends CloudWatch alarm notifications to Slack.`

`MemorySize: 128`

`Timeout: 3`

SAM – Example (2):

`AWS::TemplateFormatVersion: '2010-09-09'`

`Transform: 'AWS::Serverless-2016-10-31'`

`Description: An Amazon SNS trigger that sends CloudWatch alarm notifications to Slack.`

`Resources:`

`cloudwatchalarmtoslack:`

`Type: 'AWS::Serverless::Function'`

`Properties:`

`Handler: index.handler`

`Runtime: nodejs4.3`

`CodeUri: .`

`Description: >-`

`An Amazon SNS trigger that sends CloudWatch alarm notifications to Slack.`

`MemorySize: 128`

`Timeout: 3`

`Policies:`

`- Version: '2012-10-17'`

`Statement:`

`- Effect: Allow`

`Action:`

`- 'kms:Decrypt'`

`Resource: '*'`

`Events:`

`SNS1:`

`Type: SNS`

`Properties:`

`Topic:`

`Ref: SNSTopic1`



SAM CLI

- 1) <https://github.com/aws-labs/aws-sam-cli>
- 2) Sam init – creates new project
- 3) Local lambda debugging

AWS Cloud9 IDE

- 1) Cloud hosted IDE
- 2) Runs in the browser
- 3) Connects to any compute
- 4) Interactive terminal
- 5) Preinstalled with AWS CLI and SAM local

AWS CodeSuite

- 1) AWS CodePipeline – define release workflow
- 2) AWS CodeBuild – hosted cloud builds
- 3) AWS CodeDeploy – automated deployments
- 4) AWS CodeStar – centralized application management

AWS CloudWatch and AWS XRay

- 1) Service health monitoring
- 2) Troubleshooting
- 3) Post release operations
- 4) Instrumentation and inter-service tracing

What we will do:

- 1) Create Lambda with CodeStar
- 2) Debug/Edit it in Cloud9
- 3) Configure CodeBuild to run tests on commit
- 4) Update our Lambda Application to use XRay
- 5) Modify SAM to use CodeDeploy – safe deployments
- 6) Generate traffic – good and bad
- 7) Use CloudWatch and XRay to diagnose our live service

Let's get started!



<insert amazing awesome demo>

Thank you!

Steve Mayszak

mayszaks@amazon.com