

Building Modern Serverless Applications in .NET Core 2.0

Norm Johanson, Senior Software Engineer at AWS

Getting Setup

AWS Toolkit for Visual Studio 2017

The screenshot displays the Visual Studio 2017 interface. The main window is titled "Frontend - Microsoft Visual Studio Preview" and is in "PREVIEW" mode. The menu bar includes File, Edit, View, Project, Build, Debug, Team, Tools, Test, Analyze, Window, and Help. The toolbar shows various icons for file operations and debugging, with "Debug" and "Any CPU" selected. The status bar at the bottom indicates "Error List", "Output", and "Exception Settings".

On the left side, the "Toolbox" is visible, containing the "AWS Explorer" window. The "AWS Explorer" window shows the "Profile" set to "default" and the "Region" set to "US West (Oregon)". A list of AWS services is displayed, including Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon Elastic Container Service, Amazon RDS, Amazon S3, Amazon SimpleDB, Amazon SNS, Amazon SQS, Amazon VPC, AWS CloudFormation, AWS Elastic Beanstalk, AWS Identity and Access Management, and AWS Lambda.

The "Extensions and Updates" window is open in the center, showing a search for "AWS Toolkit". The search results list the "AWS Toolkit for Visual Studio 2017" extension, which is marked as installed with a green checkmark. The description for the extension reads: "The AWS Toolkit for Visual Studio is an extension for Microsoft Visual Studio on Windows that makes it easier for developers to develop, d...". The window also shows a list of installed extensions, including "Visual Studio Marketplace", "Controls", "Templates", "Tools", "Updates", and "Roaming Extension Manager".

At the bottom of the "Extensions and Updates" window, there is a link to "Change your Extensions and Updates settings" and a "Close" button.

Command Line Tools

Creating Projects

- **Install templates:** `dotnet new -i Amazon.Lambda.Templates`
- **Create Project:** `dotnet new lambda.S3 --region us-west-2`

Deploying Projects

- Dotnet CLI Tool **Amazon.Lambda.Tools**

```
...
| <ItemGroup>
|   <DotNetCliToolReference Include="Amazon.Lambda.Tools" Version="2.1.1" />
| </ItemGroup>
|
| </Project>
```

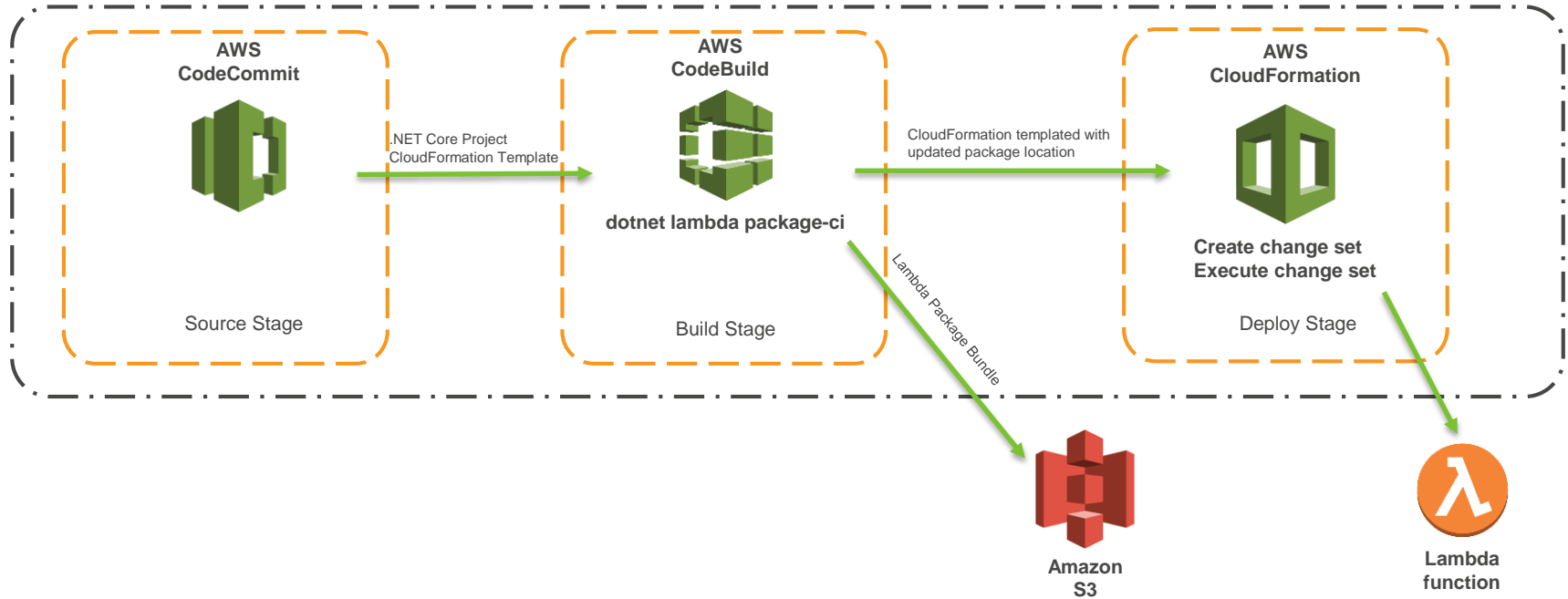
- **Deploy Command:** `dotnet lambda deploy-function`

Continuous Integrations & Continuous Delivery

AWS Code Services

- AWS CodeCommit
 - Git source control provider
- AWS CodeBuild
 - Docker based build system
- AWS CodePipeline
 - CI/CD service to connect your source, build and deployment together.
 - Supports third party source providers, like GitHub, and build systems like Jenkins.

CI - Serverless



AWS Tools for Visual Studio Team Services



AWS
CodeDeploy



AWS
CloudFormation



AWS Elastic
Beanstalk



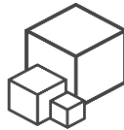
AWS
Lambda



Amazon
S3



Amazon
SNS



AWS
CLI



AWS Tools for
Windows PowerShell

Tasks for deployment:

- AWS CodeDeploy
- AWS CloudFormation
- AWS Elastic Beanstalk
- AWS Lambda

General purpose tasks:

- **Amazon S3 uploads/downloads**
- **Invoking Lambda functions**
- **Send SNS messages**
- **Run cmdlets/scripts using the AWS PowerShell Tools Module**
- **Run AWS CLI commands**

- <https://github.com/aws/aws-vsts-tools>
- <https://aws.amazon.com/vsts/>

Thank You!

Norm Johanson

Twitter: @socketnorm

AWS .NET Developer Blog: <https://aws.amazon.com/blogs/developer/category/net/>

GitHub Repos

- AWS .NET Lambda Tools: <https://github.com/aws/aws-lambda-dotnet>
- AWS Extensions for Dotnet CLI: <https://github.com/aws/aws-extensions-for-dotnet-cli>
- AWS SDK for .NET: <https://github.com/aws/aws-sdk-net>
- AWS Tools for VSTS: <https://github.com/aws/aws-vsts-tools>

