

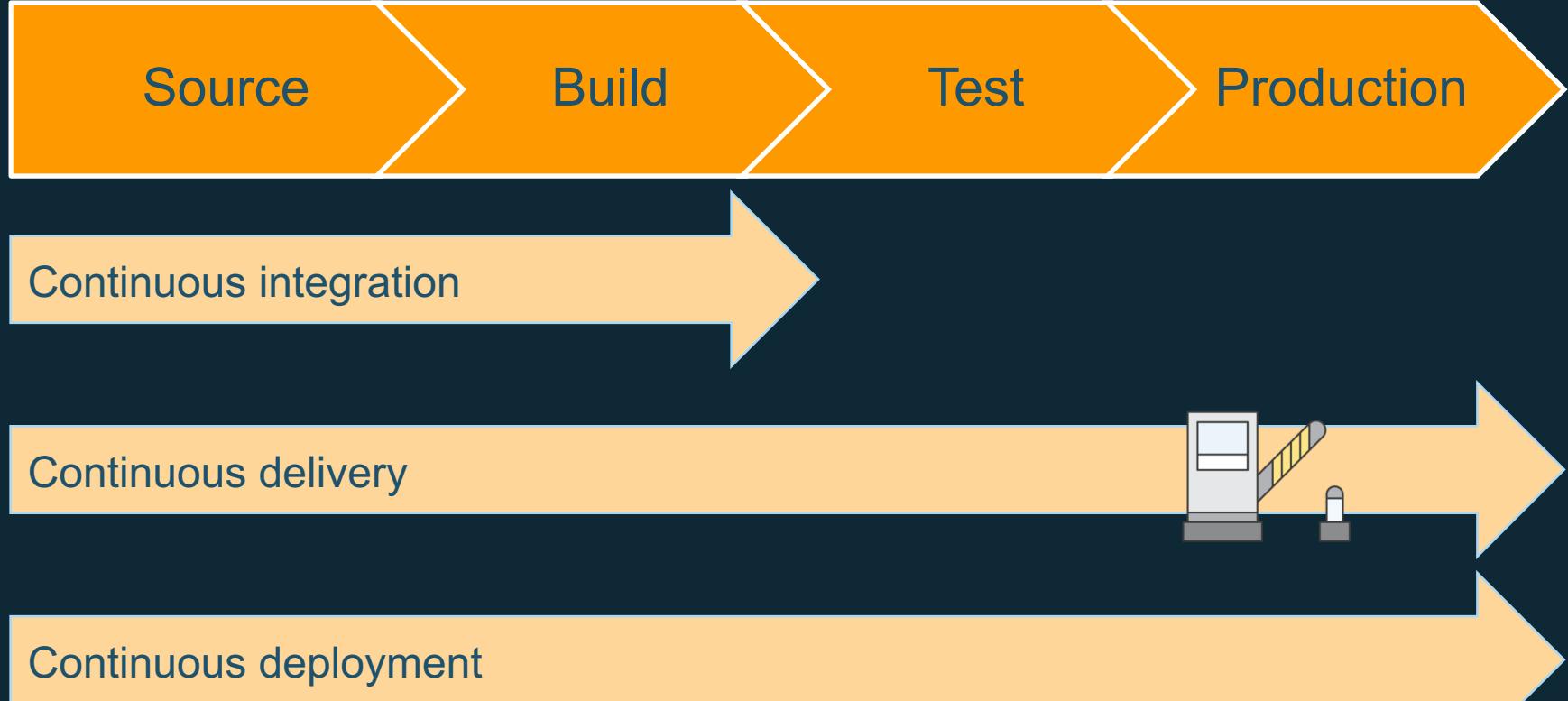


# .NET DevOps on AWS

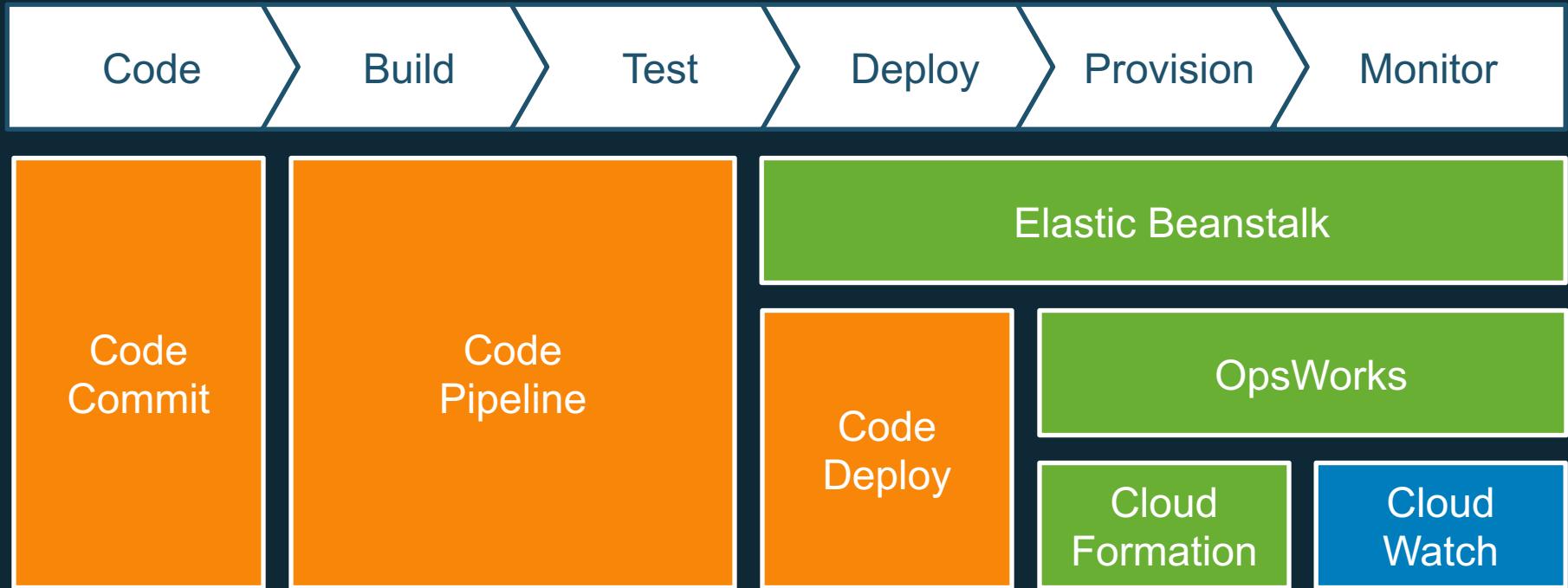
June 19, 2019



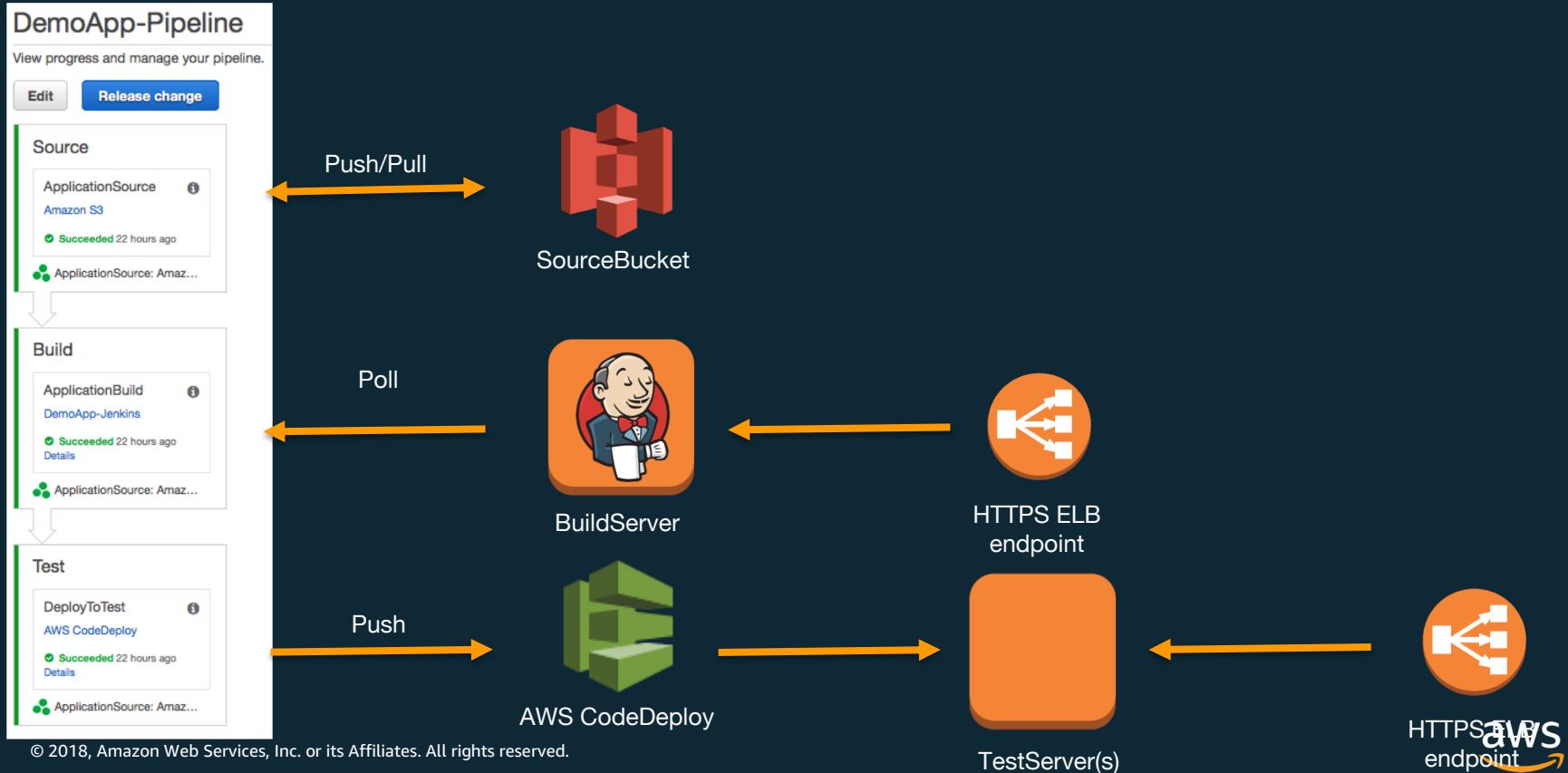
# Release processes levels



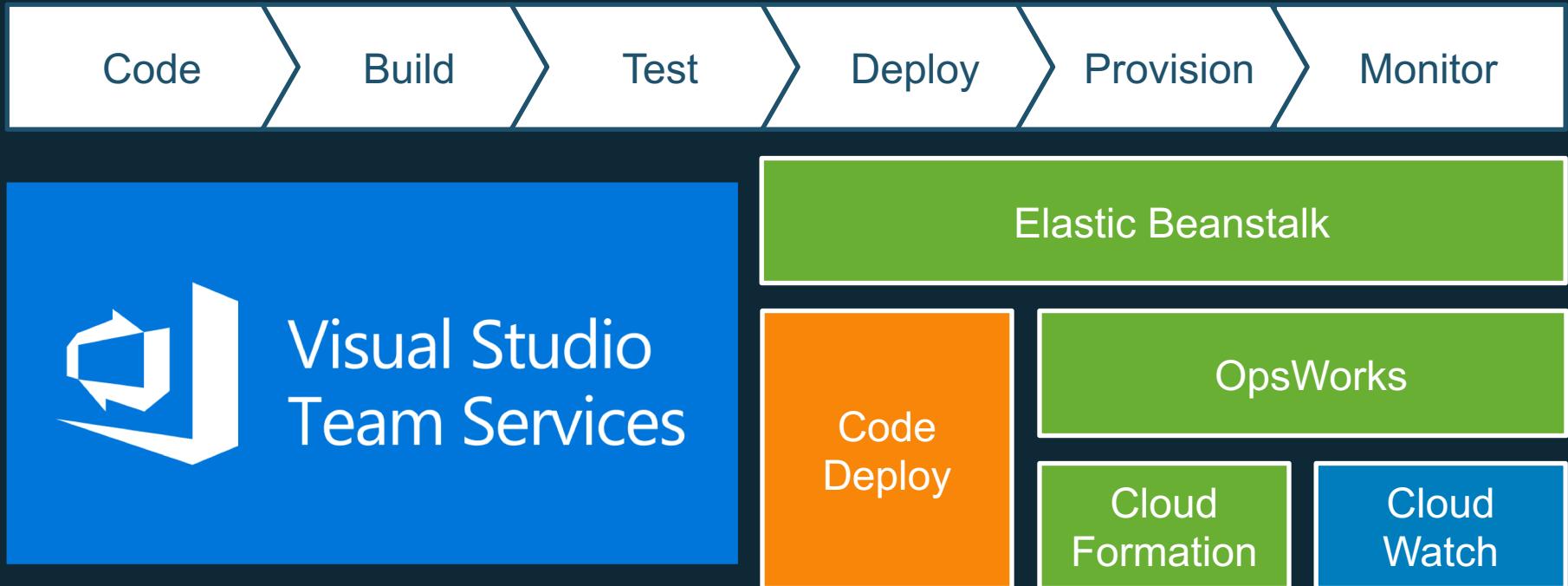
# AWS DevOps Services



# Example of a .NET CI/CD Pipeline



# AWS DevOps Services



# AWS CodeCommit

*Secure, scalable, and managed Git source control*



**git push**

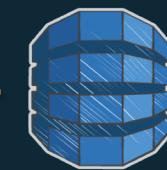
SSH or HTTPS



**CodeCommit**



Git objects  
in S3



Git index  
in DynamoDB



Encryption key  
in KMS

Data redundancy across AZs

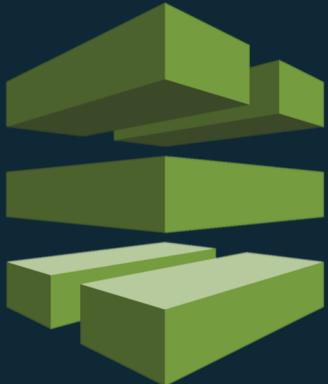
Data at rest encryption

Integrated with AWS Identity and Access Management

No repo size limit

# AWS CodePipeline

Continuous delivery service for fast and reliable application updates



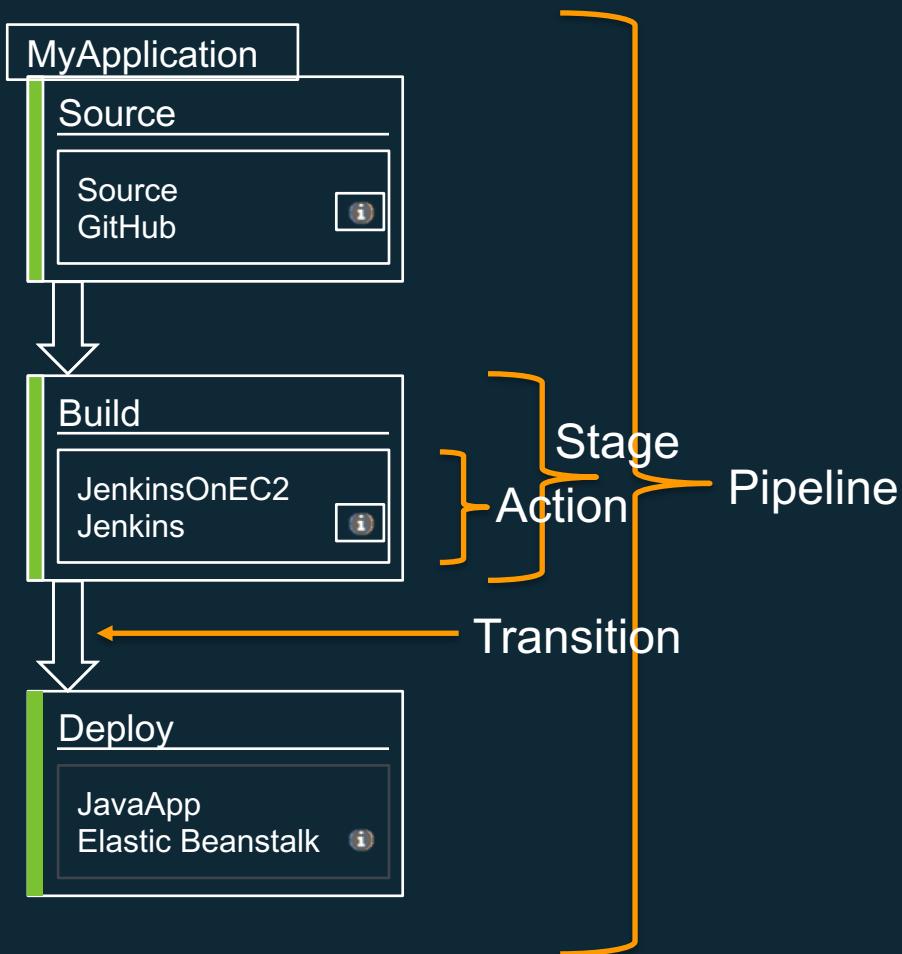
Model and visualize your software release process

Builds, tests, and deploys your code every time there is a code change

Integrates with third-party tools and AWS

# Sample Pipeline

Includes support for:  
Parallel actions  
Sequential actions  
Manual approvals



# AWS CodeDeploy

Automates code deployments to any instance

Handles the complexity of updating your applications



Avoid downtime during application deployment

Deploy to Amazon EC2 or on-premises servers, in any language and on any operating system

Integrates with third-party tools and AWS

# CodeDeploy: appspec.yml

```
version: 0.0
os: linux
files:
  - source: /octank-ticketing.war
    destination: /tmp/codedeploy-deployment-staging-area/
  #- source: /scripts/configure_http_port.xsl
  #  destination: /tmp/codedeploy-deployment-staging-area/
hooks:
  ApplicationStop:
    - location: scripts/stop_application
      timeout: 300
  BeforeInstall:
    - location: scripts/install_dependencies
      timeout: 300
  ApplicationStart:
    - location: scripts/write_codedeploy_config.sh
    - location: scripts/start_application
      timeout: 300
  ValidateService:
    - location: scripts/basic_health_check.sh
```

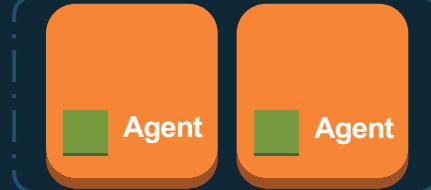
- Send application files to one directory and configuration files to another
- Set specific permissions on specific directories and files

- Remove/add instance to ELB
- Install dependency packages
- Start Apache
- Confirm successful deploy
- More!

# Choose deployment speed and group

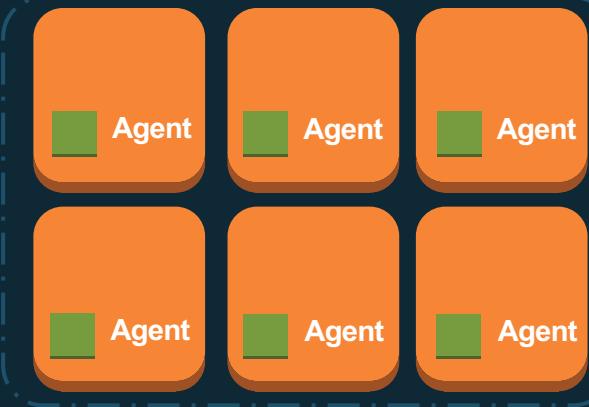


Dev deployment group



OR

Prod deployment group



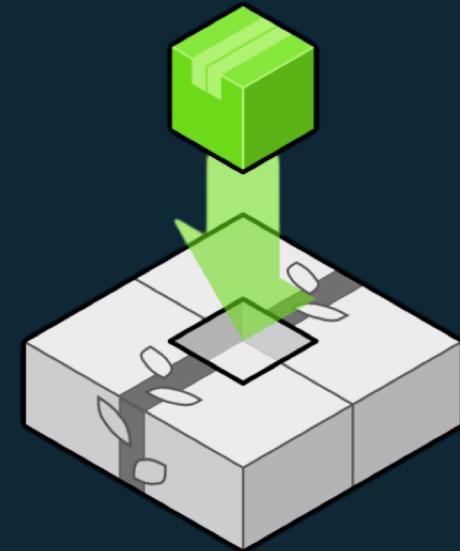
# AWS Elastic Beanstalk (EB)

Easily deploy, monitor, and scale three-tier web applications and services.

Infrastructure provisioned and managed by EB – but you maintain complete control.

Preconfigured application containers that are easily customizable.

Support for:

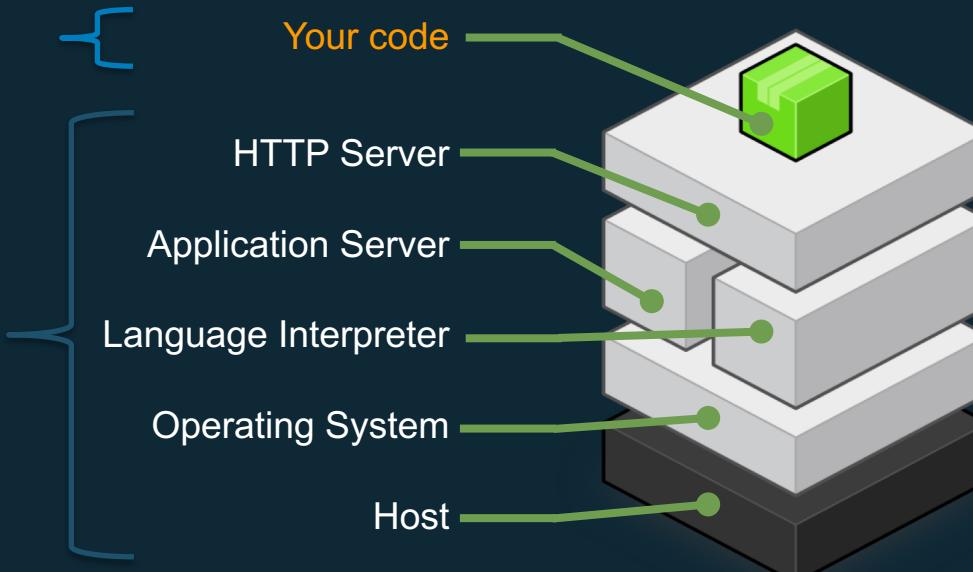




# AWS Elastic Beanstalk

Focus on building your application

Elastic Beanstalk configures each EC2 instance in your environment with the components necessary to run applications for the selected platform. No more worrying about logging into instances to install and configure your application stack.



Provided by you



Provided and managed by AWS Elastic Beanstalk (EB)

# Extensible via .ebextensions

Resources:

StartupSignupTable:

Type: AWS::DynamoDB::Table

Properties:

KeySchema:

HashKeyElement: {AttributeName: email, AttributeType: S}

ProvisionedThroughput: {ReadCapacityUnits: 1, WriteCapacityUnits: 1}

NewSignupQueue:

Type: AWS::SQS::Queue

NewSignupTopic:

Type: AWS::SNS::Topic

Properties:

Subscription:

- Endpoint:

Fn::GetOptionSetting: {DefaultValue: nobody@amazon.com, OptionName: NewSignupEmail}

Protocol: email

- Endpoint:

Fn::GetAtt: [NewSignupQueue, Arn]

Protocol: sqs

# AWS CloudFormation

Gives developers and systems administrators an easy way to create and manage a collection of related AWS resources



Understands dependencies and supports rollbacks and versioning

Allows for reusable component design strategies

Supports a portable JSON or YAML format

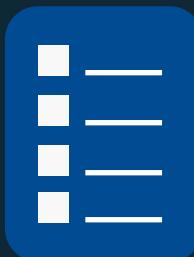
Authoring tools continue to improve

# CloudFormation – Components & Technology

Template



CloudFormation



Stack



JSON/YAML formatted file

*Parameter definition*

*Resource creation*

*Configuration actions*

Framework

*Stack creation*

*Stack updates*

*Error detection and rollback*

Configured AWS resources

*Comprehensive service support*

*Service event aware*

*Customizable*

# CloudFormation Template Anatomy

---

AwSTemplateFormatVersion: "version date"

Description:

*String*

Metadata:

*template metadata*

Parameters:

*set of parameters*

Mappings:

*set of mappings*

Conditions:

*set of conditions*

Transform:

*set of transforms*

Resources:

*set of resources*

Outputs:

*set of outputs*

# Windows Development Tools



Visual Studio



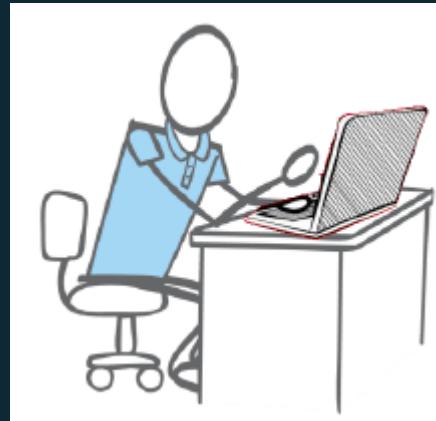
NuGet



PowerShell



Command line



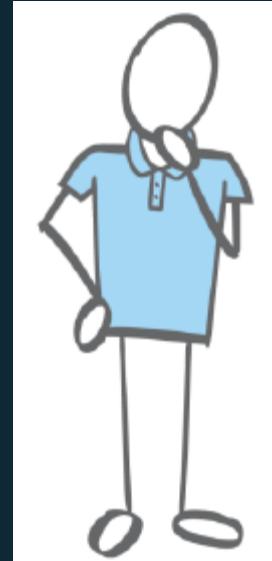
# AWS Tools for Windows



Visual Studio



AWS Toolkit for  
Visual Studio



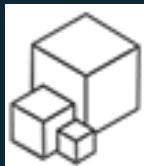
NuGet



AWS SDK for  
.NET



PowerShell



AWS Tools for  
Powershell



Command line



AWS CLI

# SDKs



Android



iOS



Java



JavaScript



.NET



Node.js



PHP



Python (boto)



Ruby



Xamarin



AWS CLI



AWS Toolkit  
for Eclipse



AWS Toolkit  
for Visual  
Studio



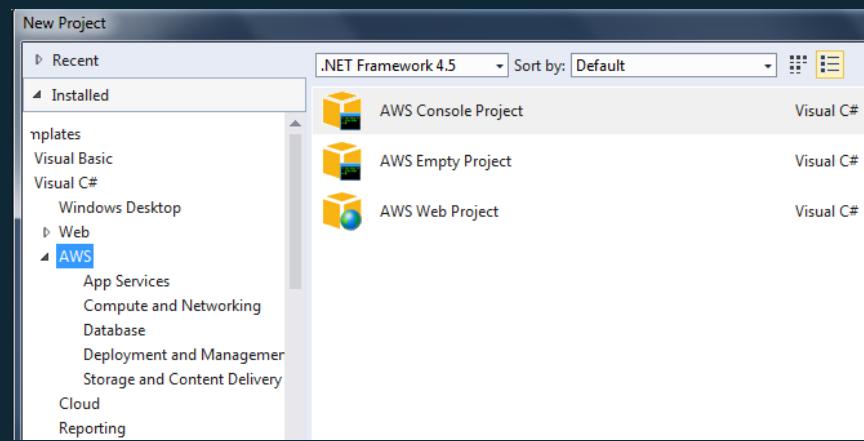
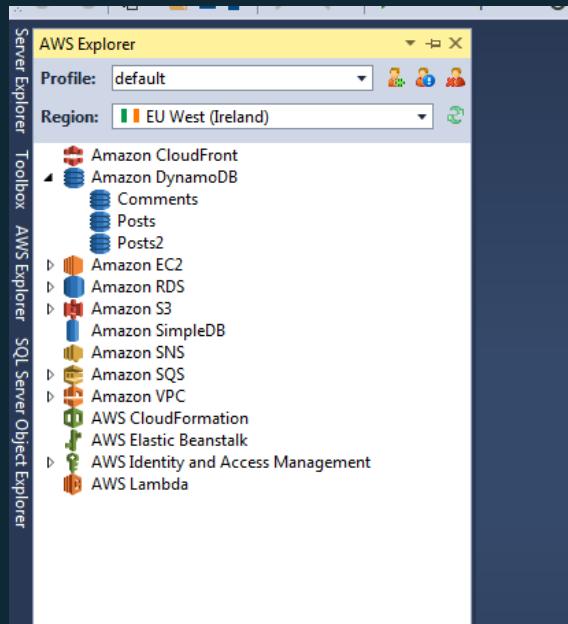
AWS Tools  
for Windows  
PowerShell

# AWS Toolkit for Visual Studio



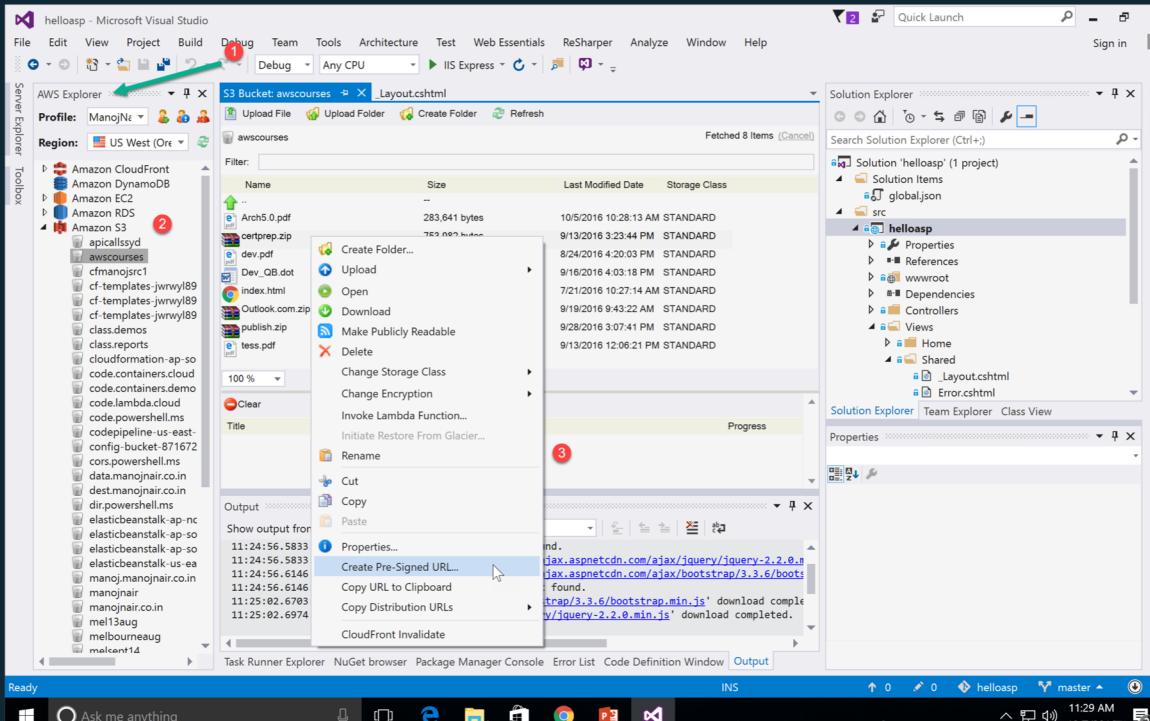
AWS Toolkit  
for Visual  
Studio

## Full integration in Visual Studio

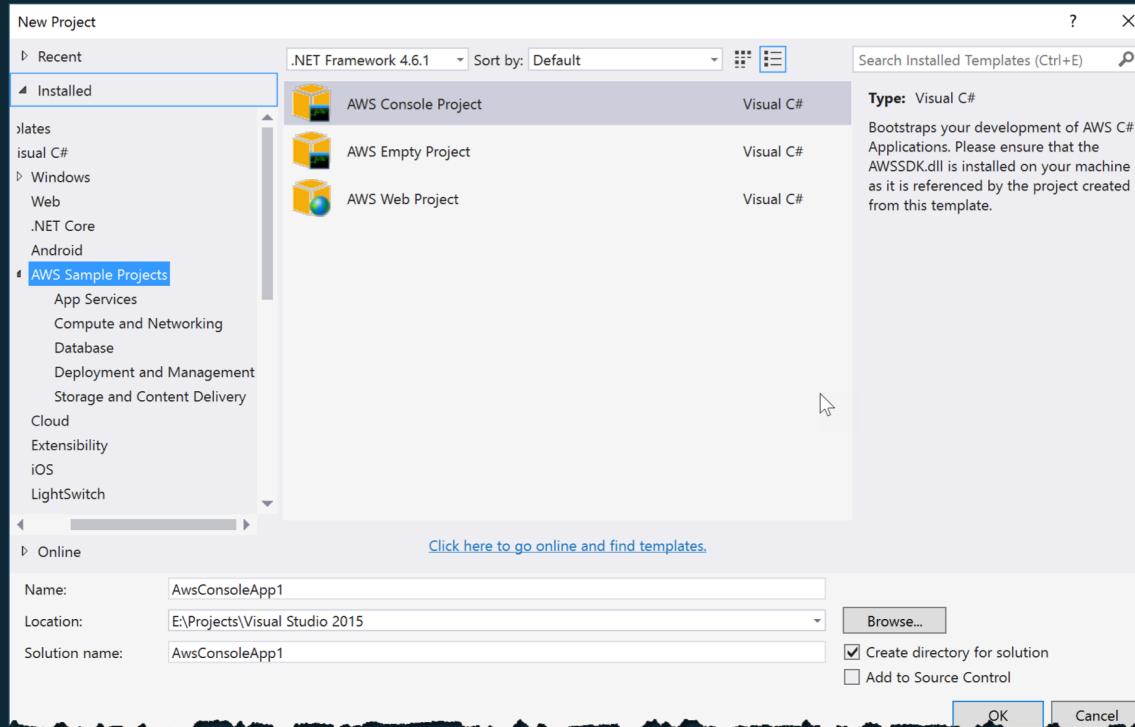


.NET SDK

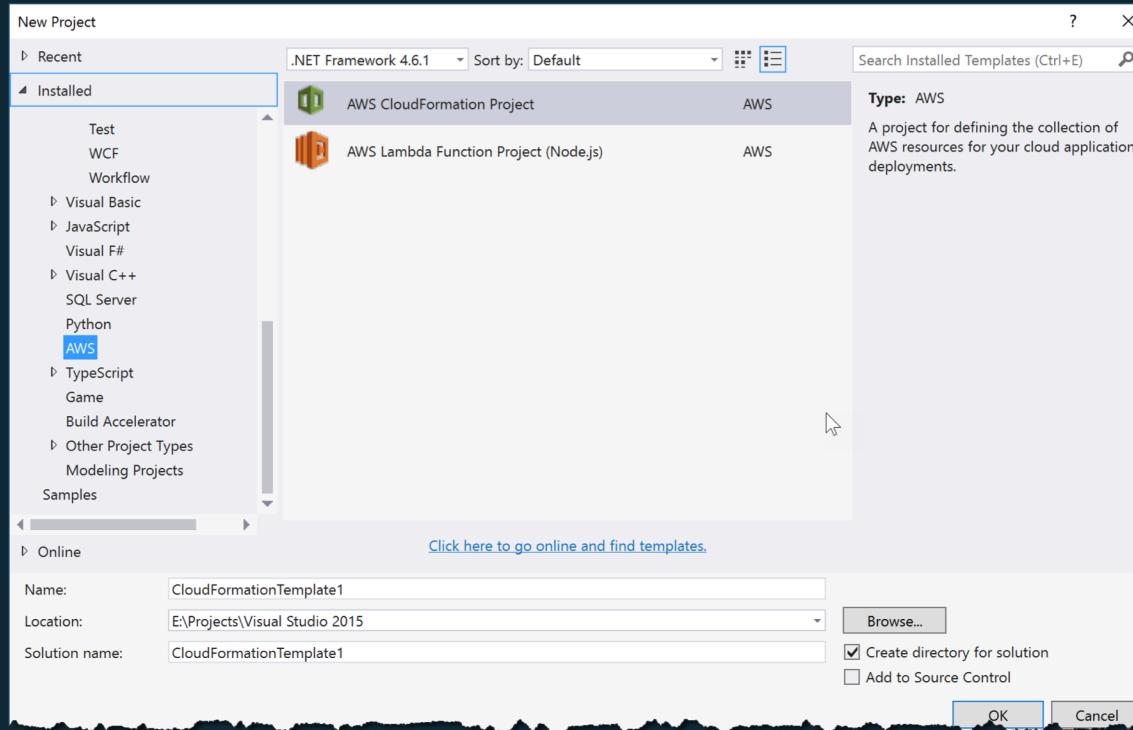
# AWS Toolkit for Visual Studio: .NET SDK



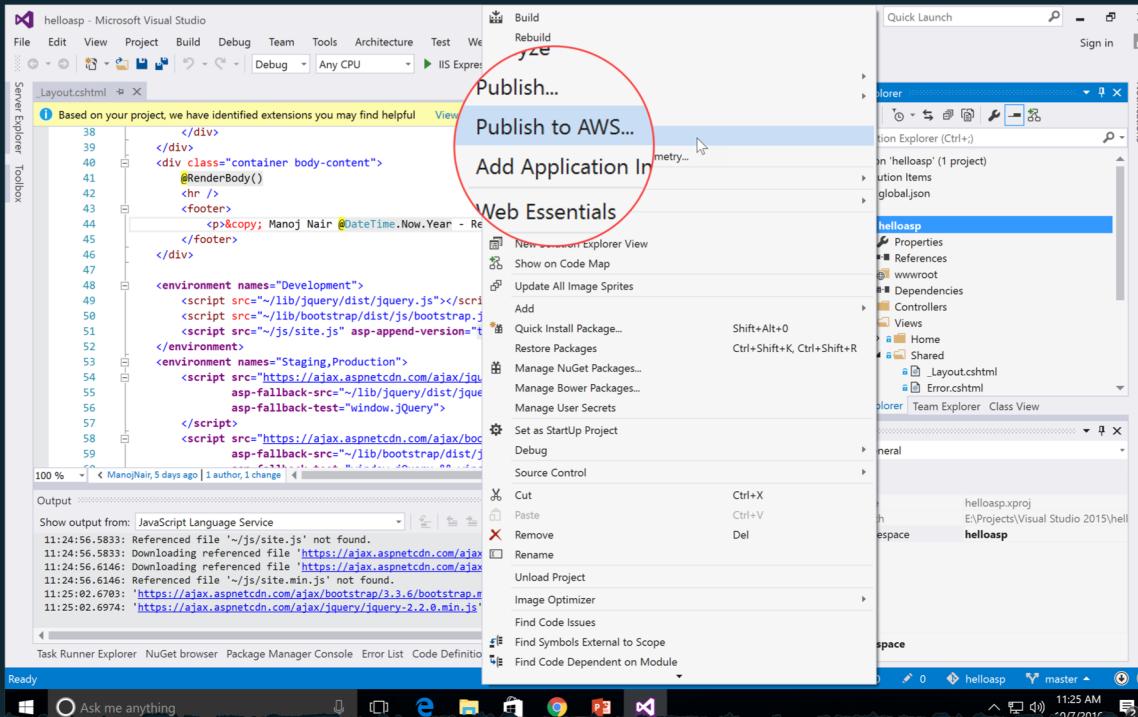
# AWS Toolkit: Starter Templates



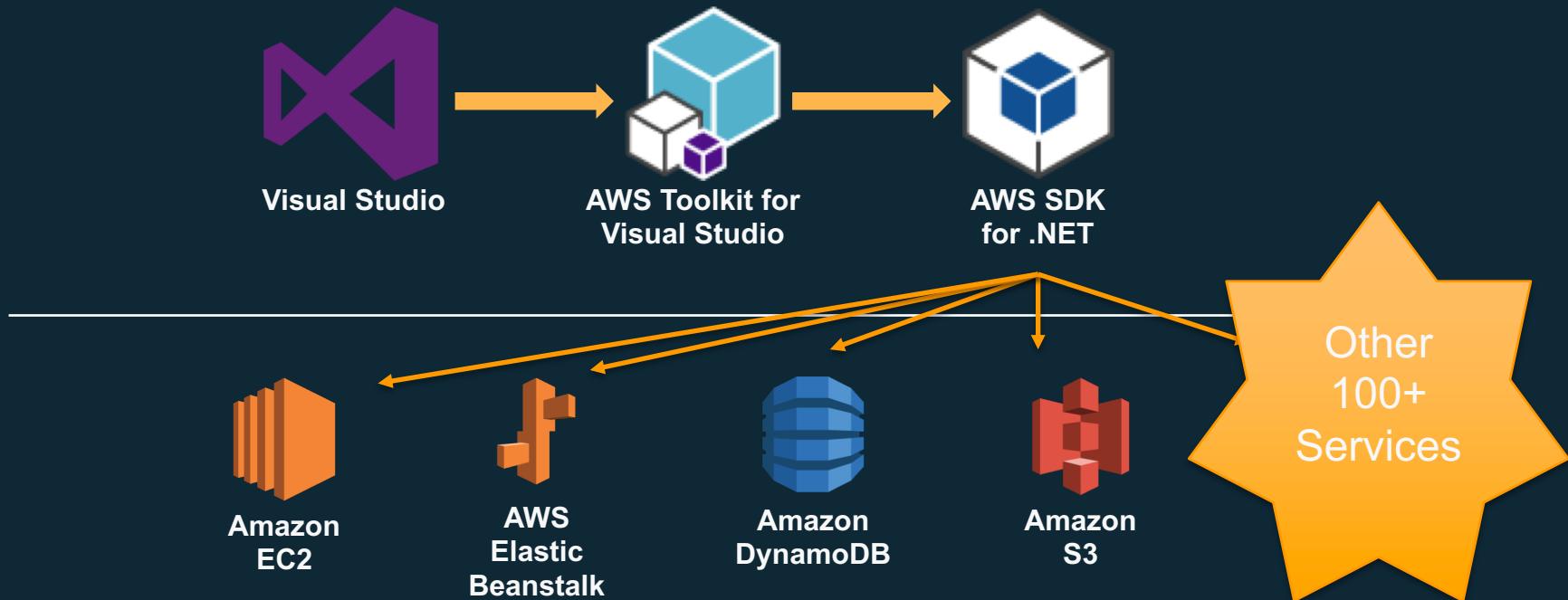
# AWS Toolkit: Boilerplate Templates



# AWS Toolkit: Elastic Beanstalk



# AWS Toolkit for Visual Studio



# AWS Toolkit: Advanced Integration

## AWS Elastic Beanstalk

- Deploy from within Visual Studio/automatic log rotation to Amazon S3



## AWS CodeCommit/CodePipeline/CodeDeploy

- Manage a large fleet (on-premises and cloud-based)



## .NET SDK and PowerShell cmdlets

- Integration in custom build pipelines in TFS or CruiseControl.NET



## AWS native integrations

- Jenkins, Bamboo have native integration to AWS
- Other IDE support AWS (Unity, Xamarin Studio, Eclipse...)

