



Masterclass

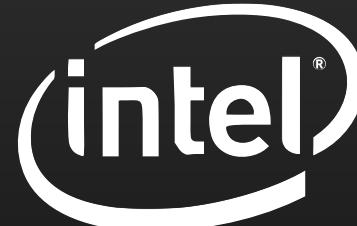
AWS CloudFormation



Ian Massingham — Technical Evangelist

✉ ianmas@amazon.com

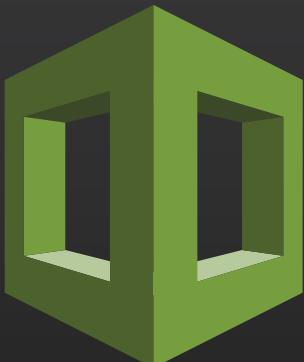
🐦 @lanMmmm



Masterclass

- 1 A technical deep dive that goes beyond the basics
- 2 Intended to educate you on how to get the best from AWS services
- 3 Show you how things work and how to get things done

AWS CloudFormation

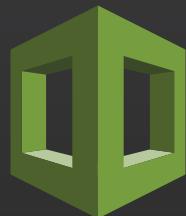


- An easy way to create & manage a collection of AWS resources
- Allows orderly and predictable provisioning and updating of resources
 - Allows you to version control your AWS infrastructure
- Deploy and update stacks using console, command line or API
 - You only pay for the resources you create

Transparent and Open

Don't reinvent the wheel

Declarative & Flexible



AWS CloudFormation

No Extra Charge

Customized via Parameters

Integration Ready

CloudFormation - Components & Technology

Template



JSON formatted file

Parameter definition

Resource creation

Configuration actions

CloudFormation



Framework

Stack creation

Stack updates

Error detection and rollback

Stack



Configured AWS services

Comprehensive service support

Service event aware

Customisable

Agenda



Creating Templates
Using a Template to Create and Manage a Stack
Working with the CloudFormation API
Working with AWS Resources
Bootstrapping Applications and Handling Updates

CREATING TEMPLATES

Manage Relationships

Reusable

Familiar JSON Format

Provide Feedback

CLOUDFORMATION TEMPLATES

Automate Generation

Avoid Collisions

Look Up Resources

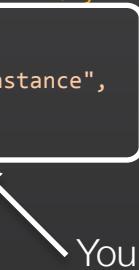
Write & Go

High Level Template Structure

```
{  
    "Description" : "A text description for the template usage",  
    "Parameters": {  
        // A set of inputs used to customize the template per deployment  
    },  
    "Resources" : {  
        // The set of AWS resources and relationships between them  
    },  
    "Outputs" : {  
        // A set of values to be made visible to the stack creator  
    },  
    "AWSTemplateFormatVersion" : "2010-09-09"  
}
```

A Simple Template that creates an EC2 Instance

```
{  
    "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",  
    "Parameters": {  
        "KeyPair": {  
            "Description": "The EC2 Key Pair to allow SSH access to the instance",  
            "Type": "String"  
        }  
    },  
    "Resources": {  
        "Ec2Instance": {  
            "Properties": {  
                "ImageId": "ami-9d23aeea",  
                "InstanceType" : "m3.medium",  
                "KeyName": {  
                    "Ref": "KeyPair"  
                }  
            },  
            "Type": "AWS::EC2::Instance"  
        }  
    },  
    "Outputs": {  
        "InstanceId": {  
            "Description": "The InstanceId of the newly created EC2 instance",  
            "Value": {  
                "Ref": "Ec2Instance"  
            }  
        }  
    },  
    "AWSTemplateFormatVersion": "2010-09-09"  
}
```



You will be asked to enter values for these parameters when you create your stack

A Simple Template that creates an EC2 Instance

```
{  
  "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",  
  "Parameters": {  
    "KeyPair": {  
      "Description": "The EC2 Key Pair to allow SSH access to the instance",  
      "Type": "String"  
    }  
  },  
  "Resources": {  
    "Ec2Instance": {  
      "Properties": {  
        "ImageId": "ami-9d23aeea",  
        "InstanceType" : "m3.medium",  
        "KeyName": {  
          "Ref": "KeyPair"  
        }  
      },  
      "Type": "AWS::EC2::Instance"  
    },  
  },  
  "Outputs": {  
    "InstanceId": {  
      "Description": "The InstanceId of the newly created EC2 instance",  
      "Value": {  
        "Ref": "Ec2Instance"  
      }  
    }  
  },  
  "AWSTemplateFormatVersion": "2010-09-09"  
}
```

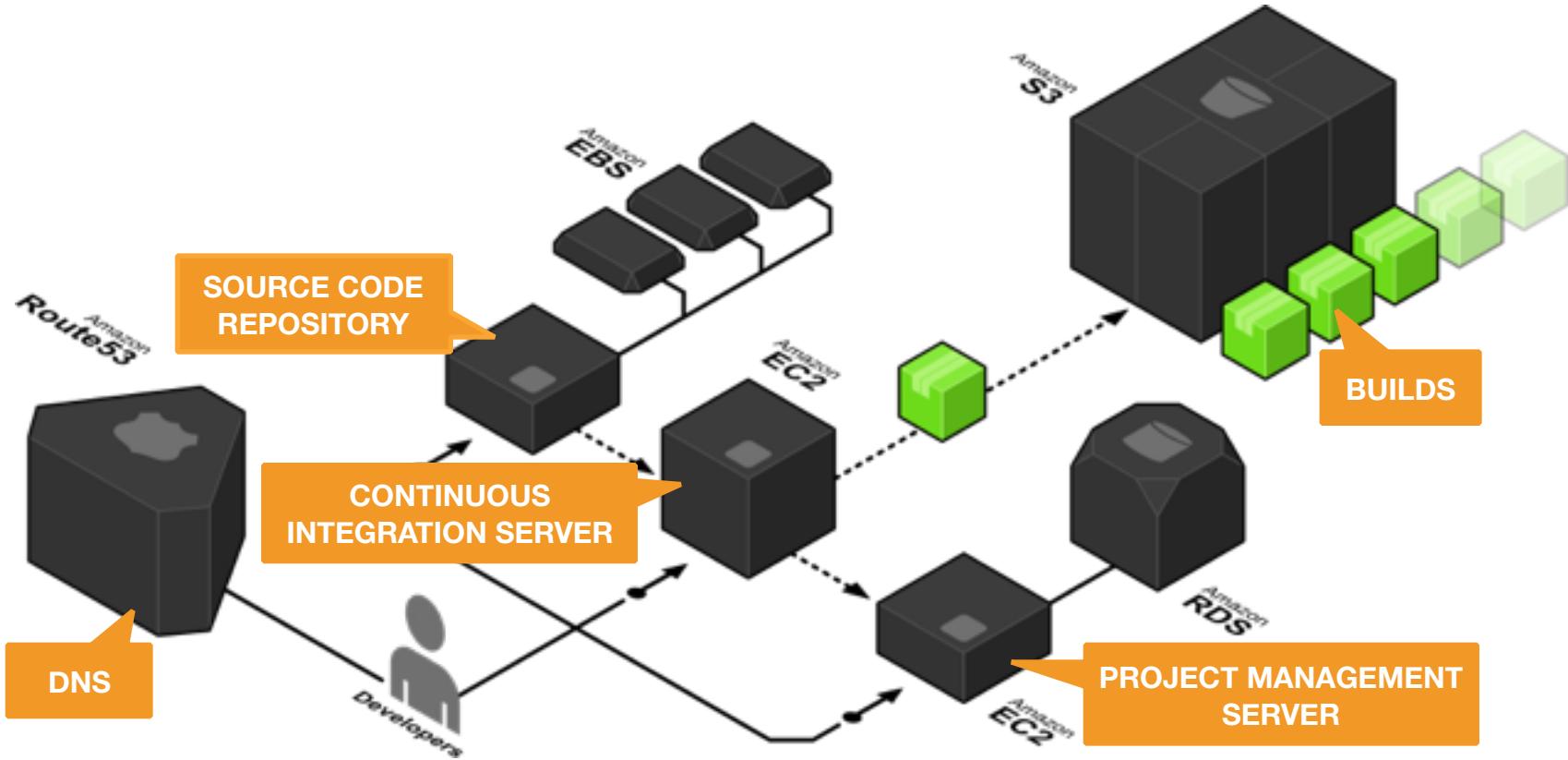


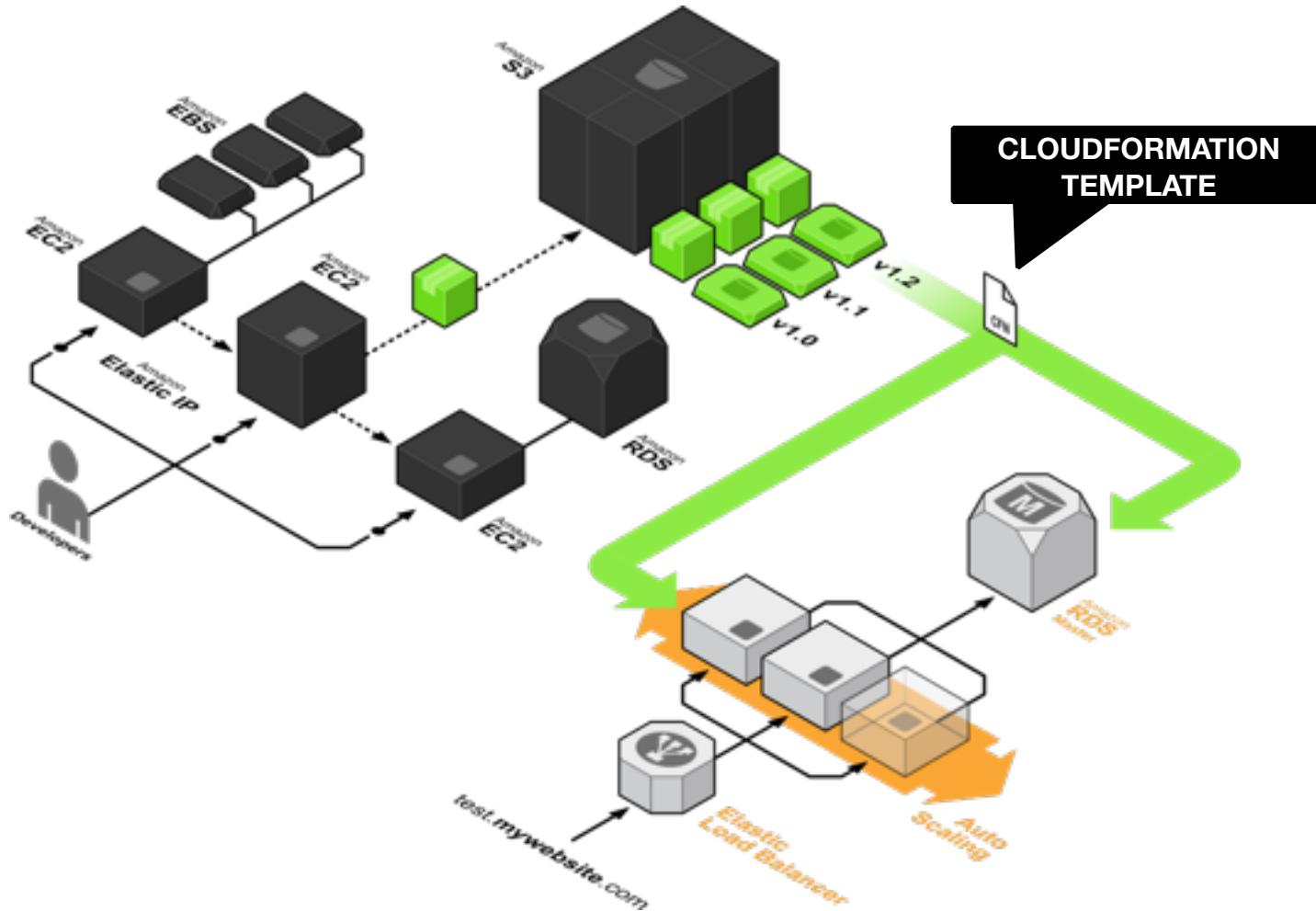
Includes statically defined properties (ImageID & Instance Type) plus a reference to the KeyPair parameter. ImageID is the AMI specific to the region that you will launch this stack in, in this case the eu-west-1 region

A Simple Template that creates an EC2 Instance

```
{  
    "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",  
    "Parameters": {  
        "KeyPair": {  
            "Description": "The EC2 Key Pair to allow SSH access to the instance",  
            "Type": "String"  
        }  
    },  
    "Resources": {  
        "Ec2Instance": {  
            "Properties": {  
                "ImageId": "ami-9d23aeea",  
                "InstanceType" : "m3.medium",  
                "KeyName": {  
                    "Ref": "KeyPair"  
                }  
            },  
            "Type": "AWS::EC2::Instance"  
        }  
    },  
    "Outputs": {  
        "InstanceId": {  
            "Description": "The InstanceId of the newly created EC2 instance",  
            "Value": {  
                "Ref": "Ec2Instance"  
            }  
        }  
    },  
    "AWSTemplateFormatVersion": "2010-09-09"  
}
```

These outputs will be returned once the template has completed execution





The screenshot shows the GitHub repository page for 'cloudtools/troposphere'. The page includes a search bar, navigation links for Explore, Features, Enterprise, and Blog, and buttons for Sign up and Sign in. The repository name 'cloudtools/troposphere' is displayed, along with a star count of 1,300 and a fork count of 250. A pull request from 'markpeet' is visible, dated Feb 21, titled 'Add version/build status badges and update tools'. The repository has 2 contributors. The README.md file is shown, containing sections for 'About', 'Installation Instructions', and 'Examples'. It also includes code snippets for pip and setup.py installation, and a simple example of creating an EC2 instance using the troposphere library.

```
>>> from troposphere import Ref, Template
>>> import troposphere.ec2 as ec2
>>> t = Template()
>>> Instance = ec2.Instance("myinstance")
>>> Instance.ImageId = "ami-91245ab"
>>> Instance.InstanceType = "t1.micro"
>>> t.add_resource(Instance)
>>> troposphere.ec2.Instance object at 0x0000000000000000
>>> 
>>> r = troposphere.ec2.Resource('myresource', 'AWS::EC2::Image')
>>> r.ImageId = 'ami-91245ab'
>>> r.SourceImageId = 'ami-91245ab'
>>> r>>>
```

github.com/cloudtools/troposphere

... but remember that a CloudFormation template is just JSON, so any tool that can generate output in JSON can be used

CREATING & MANAGING STACKS

Using a template to create and manage a stack

```
1  {
2      "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",
3      "Parameters": {
4          "KeyPair": {
5              "Description": "The EC2 Key Pair to allow SSH access to the instance",
6              "Type": "String"
7          }
8      },
9      "Resources": {
10         "Ec2Instance": {
11             "Properties": {
12                 "ImageId": "ami-dd925baa",
13                 "InstanceType": "m3.medium",
14                 "KeyName": {
15                     "Ref": "KeyPair"
16                 }
17             },
18             "Type": "AWS::EC2::Instance"
19         }
20     },
21     "Outputs": {
22         "InstanceId": {
23             "Description": "The InstanceId of the newly created EC2 instance",
24             "Value": {
25                 "Ref": "Ec2Instance"
26             }
27         }
28     },
29     "AWSTemplateFormatVersion": "2010-09-09"
30 }
31
```

Using a template to create and manage a stack

AWS Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Amazon Web Services

Compute & Networking

- Direct Connect
- EC2
- Route 53
- VPC
- WorkSpaces

Database

- DynamoDB
- ElastiCache
- RDS
- Redshift

Analytics

- Data Pipeline
- Elastic MapReduce
- Kinesis

Storage & Content Delivery

- CloudFront
- Glacier
- S3
- Storage Gateway

Deployment & Management

- CloudFormation
- CloudTrail
- CloudWatch
- Elastic Beanstalk
- IAM
- OpsWorks

App Services

- AppStream
- CloudSearch
- Elastic Transcoder
- SES
- SNS
- SQS
- SWF

Additional Resources

Getting Started

Trusted Advisor

Service Health

All services operating normally.

Updated Jun 23 2014 12:48:00
GMT+07:00

Service Health Dashboard

Set Start Page

Console Home

AWS Marketplace

Find & buy software, launch with 1-Click and play by the hour.

Feedback

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Feedback

Using a template to create and manage a stack

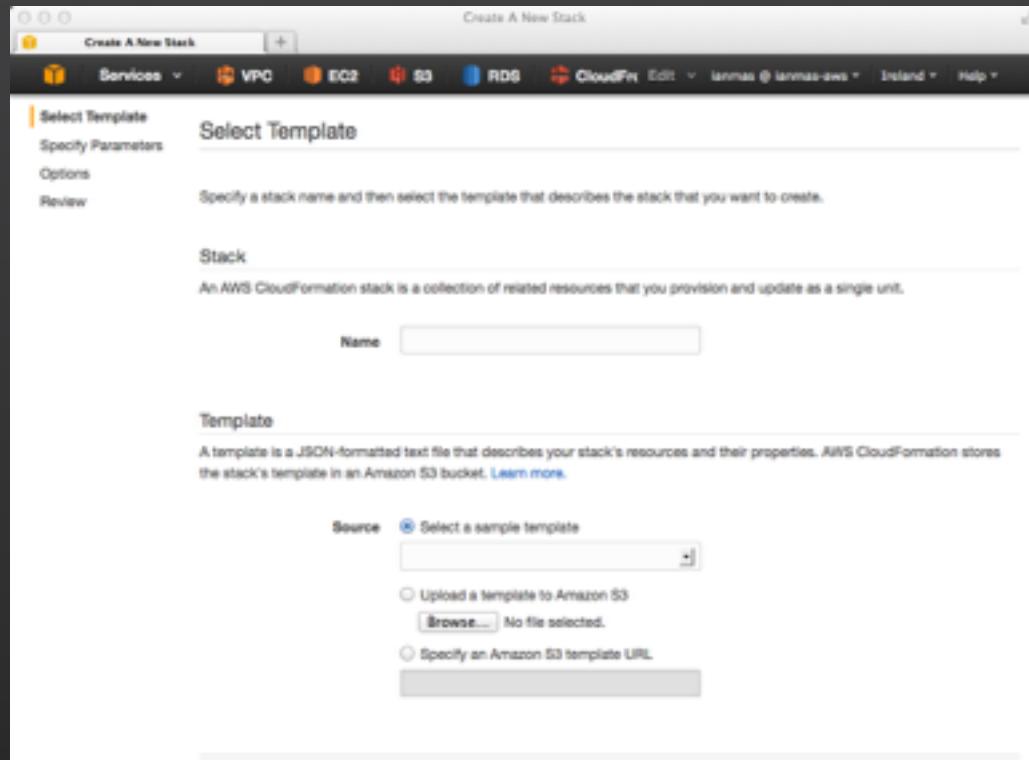
The screenshot shows the AWS CloudFormation Management Console. At the top, there's a navigation bar with links for Services (VPC, EC2, S3, RDS), CloudFront, and other account details. Below the navigation, there's a search bar labeled 'Actions' and a filter dropdown set to 'Active'. A message at the top right says 'Showing 0 stacks'.

The main content area has two sections:

- Create a Stack**: This section contains a brief introduction about CloudFormation and its capabilities. It states: "AWS CloudFormation allows you to quickly and easily deploy your infrastructure resources and applications on AWS. You can use one of the templates we provide to get started quickly with applications like WordPress or Drupal, one of the many sample templates or create your own template." Below this, it says: "You do not currently have any stacks. Click the 'Create New Stack' button below to create a new AWS CloudFormation Stack." A blue button labeled "Create New Stack" is highlighted with a red rectangular box.
- Create a Template from your Existing Resources**: This section provides instructions for creating a template from existing resources using the CloudFormer tool. It says: "If you already have AWS resources running, the CloudFormer tool can create a template from your existing resources. This means you can capture and redeploy applications you already have running." It then describes the process: "To do this, click Launch CloudFormer and create an AWS CloudFormation stack that runs the CloudFormer tool. After the stack creation is complete, navigate to the CloudFormer URL available on the Outputs tab." A blue button labeled "Launch CloudFormer" is visible.

At the bottom of the page, there's a footer with copyright information: "© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved." and links to "Privacy Policy" and "Terms of Use". There are also "Feedback" and "Help" buttons.

Using a template to create and manage a stack



Using a template to create and manage a stack

Create A New Stack

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Select Template

Specify Parameters

Options

Review

Stack

An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.

Name

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more](#).

Source Select a sample template Upload a template to Amazon S3 Specify an Amazon S3 template URL

Browse... No file selected.

[Cancel](#) [Next](#)

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)

Using a template to create and manage a stack

The screenshot shows the 'Create A New Stack' wizard in the AWS CloudFormation console. The top navigation bar includes 'Services' (with VPC, EC2, S3, RDS, CloudFront), 'Edit', and user information ('ianmas @ ianmas-aws', 'Ireland', 'Help'). The left sidebar has tabs: 'Select Template' (highlighted in orange), 'Specify Parameters', 'Options', and 'Review'. The main content area is titled 'Select Template' with the sub-section 'Stack'. It explains what a CloudFormation stack is and provides a text input field for the stack name, which is set to 'ec2InstanceDemoStack'. Below this is the 'Template' section, which defines a template as a JSON-formatted text file describing resources and properties. It mentions that the stack's template is stored in an Amazon S3 bucket and provides a link to 'Learn more'. There are three options for specifying the template source: 'Select a sample template' (radio button), 'Upload a template to Amazon S3' (radio button, selected), and 'Specify an Amazon S3 template URL'. The 'Upload a template to Amazon S3' field contains the text 'Browse... | ec2Instance.template', with the file path highlighted by a red rectangle. At the bottom right are 'Cancel' and 'Next' buttons, and a 'Feedback' link at the very bottom.

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with tabs for Services (selected), VPC, EC2, S3, RDS, CloudFront, Edit, IAM user (ianmas), Ireland, and Help. The main title is "Create A New Stack". On the left, a vertical sidebar lists steps: "Select Template" (disabled), "Specify Parameters" (selected), "Options", and "Review". The main content area is titled "Specify Parameters" with the sub-instruction "Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template." Below this is a "Parameters" section containing a "KeyPair" input field with a placeholder "The EC2 Key Pair to allow SSH access to the instance". At the bottom right of the main area are "Cancel", "Previous", and "Next" buttons. The footer contains copyright information: "© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved." followed by links to "Privacy Policy" and "Terms of Use", and a "Feedback" button.

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with tabs for Services (selected), VPC, EC2, S3, RDS, CloudFront, Edit, IAM user (ianmas), Ireland, and Help. The main title is "Create A New Stack". On the left, a vertical sidebar lists steps: "Select Template" (disabled), "Specify Parameters" (selected), "Options", and "Review". The main content area is titled "Specify Parameters" with the sub-section "Parameters". It shows a parameter named "ManagementKeyPair" which is highlighted with an orange border. A tooltip for this parameter states: "The EC2 Key Pair to allow SSH access to the instance". Below the parameters, there are "Cancel", "Previous", and "Next" buttons. At the bottom of the page, there's a footer with copyright information: "© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved." followed by links to "Privacy Policy" and "Terms of Use", and a "Feedback" button.

Using a template to create and manage a stack

Create A New Stack

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Select Template
Specify Parameters
Options
Review

Options

Tags

You can specify tags (key-value pairs) for resources in your stack. You can add up to 10 unique key-value pairs for each stack. [Learn more.](#)

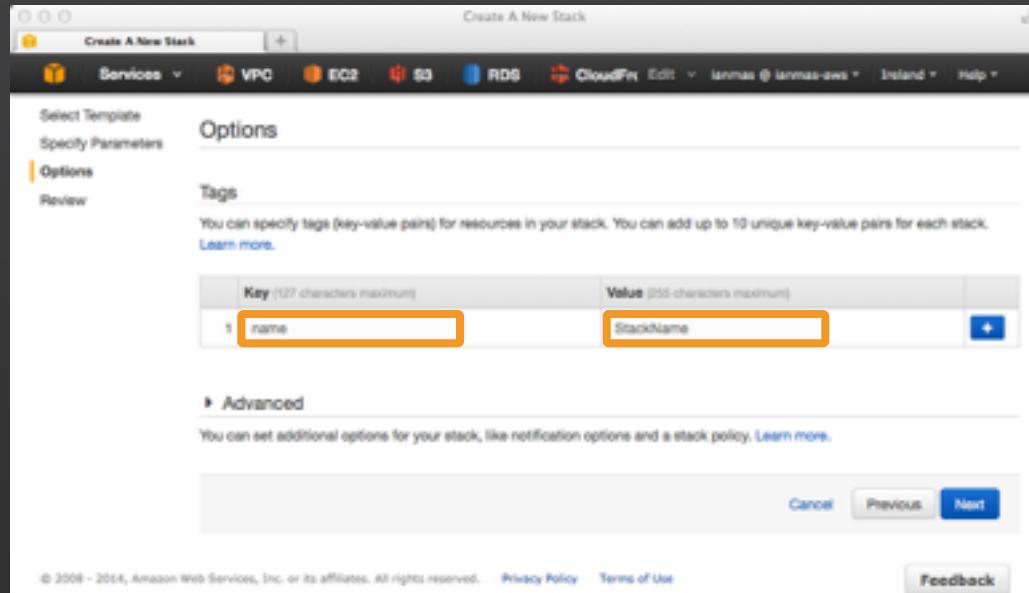
Key (1-27 characters maximum)	Value (55 characters maximum)
1 name	StackName

Advanced

You can set additional options for your stack, like notification options and a stack policy. [Learn more.](#)

Cancel Previous Next

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)



Using a template to create and manage a stack

Create A New Stack

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Select Template

Specify Parameters

Options

Review

Template

Name: ec2InstanceDemoStack
Template URL: <https://s3.amazonaws.com/ctf-templates-tfheiyvdrbr-eu-west-1/20141740-ec2instance.template>
Description: Create an EC2 instance running the newest Amazon Linux AMI.
Estimate cost: Link is not available

Parameters

KeyPair: ManagementKeyPair
Create IAM resources: False

Options

Tags

name: StackName

Advanced

Notification: none
Timeout: none
Rollback on failure: Yes

Cancel Previous Create

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Actions

Filter: Active - By Name:

Showing 1 stack

Stack Name	Created Time	Status	Description
ec2InstanceDemoStack	2014-06-23 12:59:47 UTC+0100	CREATE_COMPLETE	Create an EC2 instance running the newest Amazon

Events

4 more events available to display.

2014-06-23	Status	Type	Logical ID	Status Reason
12:59:47 UTC+0100	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	User Initiated

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Actions

Filter: Active - By Name:

Showing 1 stack

Stack Name	Created Time	Status	Description
ec2instanceDemoStack	2014-06-23 12:59:47 UTC+0100	CREATE_COMPLETE	Create an EC2 instance running the newest Amazon Linux AMI.

Events

Date	Status	Type	Logical ID	Status Reason
2014-06-23 12:00:43 UTC+0100	CREATE_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
2014-06-23 12:00:42 UTC+0100	CREATE_COMPLETE	AWS::EC2::Instance	Ec2instance	
2014-06-23 12:00:42 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::Instance	Ec2instance	Resource creation initiated
2014-06-23 12:59:52 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::Instance	Ec2instance	
2014-06-23 12:59:47 UTC+0100	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	User initiated

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Actions

Filter: Active - By Name:

Showing 1 stack

Stack Name	Created Time	Status	Description
ec2InstanceDemoStack	2014-06-23 12:59:47 UTC+0100	CREATE_COMPLETE	Create an EC2 instance running the newest Amazon

Events

Overview	Status	Type	Logical ID	Status Reason
Outputs	CREATE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceDemoStack	
Resources	CREATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
Events	CREATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Resource creation initiated
Template				
Parameters	CREATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	User Initiated
Tags	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	User Initiated
Stack Policy				

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Ireland Help

Actions

Filter: Active - By Name:

Showing 1 stack

Stack Name	Created Time	Status	Description
ec2InstanceDemoStack	2014-06-23 12:59:47 UTC+0100	CREATE_COMPLETE	Create an EC2 instance running the newest Amazon

Outputs

Key	Value	Description
InstanceId	i-d2a60792	The InstanceId of the newly created EC2 instance

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Feedback

Using a template to create and manage a stack

Incorrect Syntax

```
1  "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",  
2  "Parameters": {  
3      "KeyPair": {  
4          "Description": "The EC2 Key Pair to allow SSH access to the instance",  
5          "Type": "String"  
6      }  
7  },  
8  "Resources": {  
9      "Ec2Instance": {  
10         "Properties": {  
11             "ImageId": "ami-0d925baa",  
12             "InstanceType": "m3.medium",  
13             "KeyName": {  
14                 "Ref": "KeyPair"  
15             },  
16             "SecurityGroups": "ssh-tools"  
17         },  
18         "Type": "AWS::EC2::Instance"  
19     }  
20 },  
21 "Outputs": {  
22     "InstanceId": {  
23         "Description": "The InstanceId of the newly created EC2 instance",  
24         "Value": {  
25             "Ref": "Ec2Instance"  
26         }  
27     }  
28 },  
29 "AMSTemplateFormatVersion": "2018-09-09"  
30 }  
31  
32 }  
33 }  
34 }  
35 }  
36 .."AWS::ServerFarm::LambdaFunctionMapping": "5010-00-00-00",  
37 }  
38 }  
39 }  
40 }
```

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Inland Help

Actions Create Stack Name: Showing 1 stack

Update Stack Delete Stack

	Created Time	Status	Description
arn:aws:cloudformation:us-east-1:123456789012:stack/test-stack/12345678901234567890	2014-06-23 12:59:47 UTC+0100	CREATE_COMPLETE	Create an EC2 instance running the newest Amazon Linux AMI.

Outputs

Key	Value	Description
InstanceId	i-d2a60792	The InstanceId of the newly created EC2 instance.

© 2006 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Feedback

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation 'Update Stack' interface. The top navigation bar includes 'Services' (VPC, ECS, S3, RDS, CloudFront), 'Edit', 'ianmas @ ianmas-aws', 'Inland', and 'Help'. The main left sidebar has tabs: 'Select Template' (highlighted in orange), 'Specify Parameters', 'Options', and 'Review'. The 'Select Template' tab displays the following content:

Select Template

To update an existing stack, provide a template that specifies the changes for the resources and properties that you want to update. AWS CloudFormation updates only the resources that have changed. [Learn more.](#)

Stack

An AWS CloudFormation stack is a collection of related resources that you provision and update as a single unit.

Name: ec2instanceDemoStack

Template

A template is a JSON-formatted text file that describes your stack's resources and their properties. AWS CloudFormation stores the stack's template in an Amazon S3 bucket. [Learn more.](#)

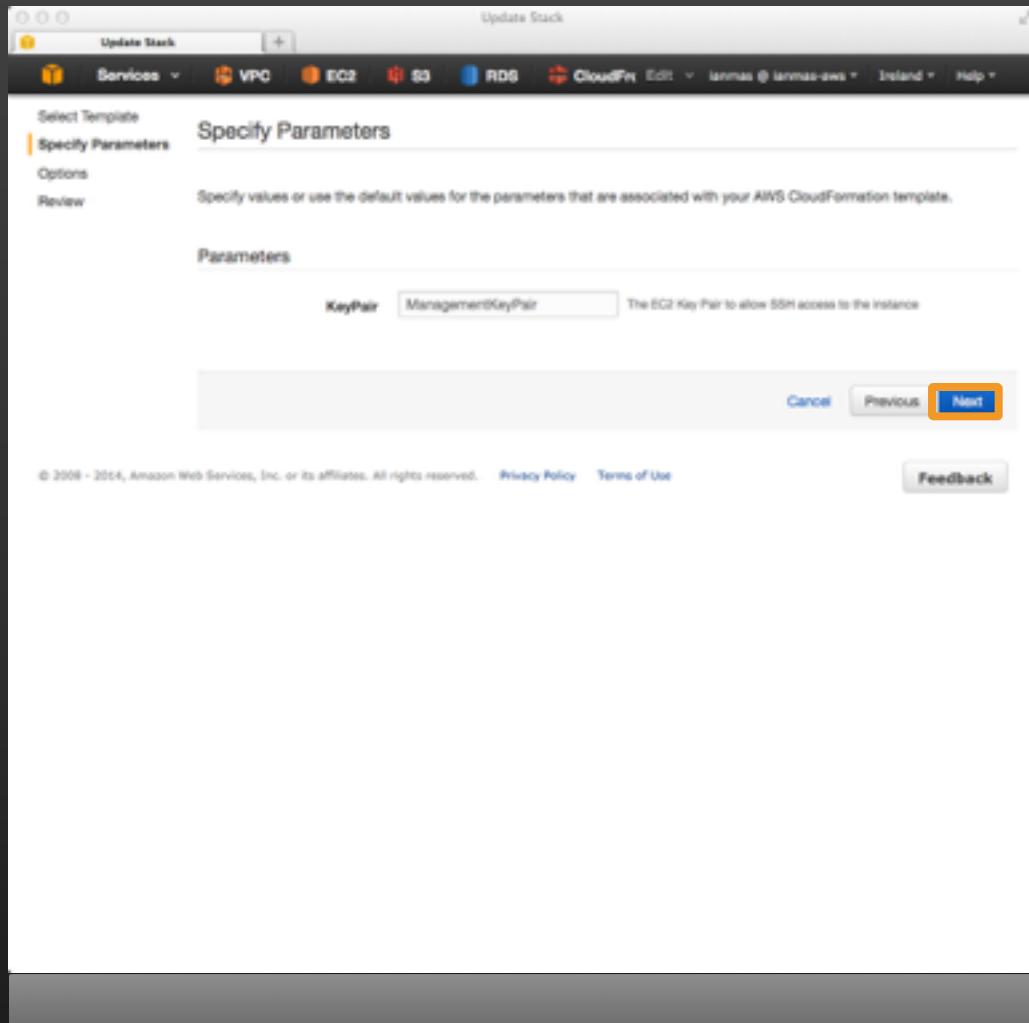
Source:

- Select a sample template
- Upload a template to Amazon S3
[Browse...](#) **ec2instance.template**
- Specify an Amazon S3 template URL
- Use existing template

Next Step

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)

Using a template to create and manage a stack



Provides Feedback

Using a template to create and manage a stack

CloudFormation Management Console

CloudFormation Management C... Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws In...

Showing 1 item

Stack Name	Created Time	Status	Description
ec2instanceDemoStack	2014-06-23 12:59:48 UTC+0100	UPDATE_ROLLBACK_COMPLETE	Create an EC2 instance running the newest Amazon Linux AMI.

Events

- 13:37:55 UTC+0100 UPDATE_ROLLBACK_COMPLETE AWS::CloudFormation::Stack ec2instanceDemoStack
- 13:37:54 UTC+0100 UPDATE_ROLLBACK_COMPLETE AWS::CloudFormation::Stack ec2instanceDemoStack
- 13:37:52 UTC+0100 UPDATE_COMPLETE AWS::EC2::Instance Ec2Instance ec2instanceDemoStack
- 13:37:27 UTC+0100 UPDATE_ROLLBACK_IN_PROGRESS AWS::CloudFormation::Stack RESOURCES

13:37:26 UTC+0100 UPDATE_FAILED AWS::EC2::Instance Ec2Instance

The following resource(s) failed to update: [Ec2Instance]. Value of property SecurityGroups must be of type List of String

Feedback

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

See Us in Seattle, DC, London, Paris, Tokyo, Sydney, Singapore, Mumbai, Bangalore, and São Paulo

Using a template to create and manage a stack

CloudFormation Management Console

CloudFormation Management C... Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws In Progress

Actions Filter: Active - By Name: Showing 1 item

Stack Name	Created Time	Status	Description
ec2instanceDemoStack	2014-06-23 12:59:48 UTC+0100	UPDATE_ROLLBACK_COMPLETE	Create an EC2 instance running the newest Amazon Linux AMI.

Events

- 13:37:55 UTC+0100 UPDATE_ROLLBACK_COMPLETE AWS::CloudFormation::Stack ec2instanceDemoStack
- 13:37:54 UTC+0100 UPDATE_ROLLBACK_COMPLETE AWS::CloudFormation::Stack ec2instanceDemoStack
- 13:37:52 UTC+0100 UPDATE_COMPLETE AWS::EC2::Instance Ec2Instance ec2instanceDemoStack
- 13:37:27 UTC+0100 UPDATE_ROLLBACK_IN_PROGRESS AWS::CloudFormation::Stack RESOURCES

13:37:26 UTC+0100 UPDATE_FAILED AWS::EC2::Instance Ec2Instance

The following resource(s) failed to update:
Ec2Instance
The value of property SecurityGroups must be of type List of String

Feedback

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

See Us in Seattle, DC, London, Paris, Tokyo, Sydney, Singapore, Mumbai, Bangalore, and São Paulo

Feedback

Provides Feedback

Using a template to create and manage a stack

Correct SYNTAX

```
1  {
2      "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",
3      "Parameters": {
4          "KeyPair": {
5              "Description": "The EC2 Key Pair to allow SSH access to the instance",
6              "Type": "String"
7          }
8      },
9      "Resources": {
10         "Ec2Instance": {
11             "Properties": {
12                 "ImageId": "ami-0d925baa",
13                 "InstanceId": "i-0d925baa",
14                 "KeyName": {
15                     "Ref": "KeyPair"
16                 },
17                 "SecurityGroups": [ "ssh-tools" ]
18             },
19             "Type": "AWS::EC2::Instance"
20         }
21     },
22     "Outputs": {
23         "InstanceId": {
24             "Description": "The InstanceId of the newly created EC2 instance",
25             "Value": {
26                 "Ref": "Ec2Instance"
27             }
28         }
29     },
30     "AWSTemplateFormatVersion": "2010-09-09"
31 }
32 }
```

Correct Syntax

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Inland Help

Actions Filter: Active By Name: Showing 1 stack

Stack Name	Created Time	Status	Description
ec2instanceDemoStack	2014-06-23 12:59:48 UTC+0100	UPDATE_COMPLETE	Create an EC2 instance running the latest Amazon Linux AMI.

Events

Date	Status	Type	Logical ID	Status Reason
2014-06-23 13:45:53 UTC+0100	UPDATE_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
2014-06-23 13:45:53 UTC+0100	DELETE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
2014-06-23 13:45:27 UTC+0100	DELETE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	
2014-06-23 13:45:25 UTC+0100	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	
2014-06-23 13:45:23 UTC+0100	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
2014-06-23 13:44:35 UTC+0100	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Resource creation initiated
2014-06-23 13:44:34 UTC+0100	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Requested update requires the creation of a new physical resource; hence creating one.
2014-06-23 13:44:27 UTC+0100	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	User Initiated
2014-06-23 13:37:55 UTC+0100	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
2014-06-23 13:37:54 UTC+0100	UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	
2014-06-23 13:37:52 UTC+0100	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	

© 2006 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Replaces Resources

Using a template to create and manage a stack

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with sections like EC2 Dashboard, Instances (which is selected and highlighted in orange), Images, Elastic Block Store, Network & Security, and Auto Scaling. The main area displays a table of instances. One instance, with the ID **i-86872bc5**, is highlighted with an orange box. Below the table, there's a detailed view for this specific instance. In the bottom right corner of this view, another orange box highlights the **ssh-tools** entry under the **Security groups** section.

Name	Instance ID	Instance Type	Availability Zone	Instance State
	i-86872bc5	m3.medium	eu-west-1a	running
	i-02a03792	m3.medium	eu-west-1c	terminated

Instance: i-86872bc5 Public DNS: ec2-54-76-124-9.eu-west-1.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID: i-86872bc5	Public DNS: ec2-54-76-124-9.eu-west-1.compute.amazonaws.com	Public IP: 54.76.124.9	Availability zone: eu-west-1a
Instance state: running	Private DNS: ip-172-31-32-42.eu-west-1.compute.internal	Elastic IP: -	Security group: ssh-tools, fw-rules
Instance type: m3.medium	Private IPs: 172.31.32.42	Scheduled events: No scheduled events	AMI ID: ami-0f3e0a2d
Secondary private IPs:	VPC ID: vpc-b7b8b3d5	AMI ID: ami-0f3e0a2d	

Using a template to create and manage a stack

CloudFormation Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Inland Help

Create Stack Name: Showing 1 stack

Delete Stack

	Created Time	Status	Description
ec2instanceDemoStack	2014-06-23 12:59:48 UTC+0100	UPDATE_COMPLETE	Create an EC2 instance running the latest Amazon Linux AMI.

Events +

2014-06-23	Status	Type	Logical ID	Status Reason
13:45:53 UTC+0100	UPDATE_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:45:53 UTC+0100	DELETE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
13:45:27 UTC+0100	DELETE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	
13:45:25 UTC+0100	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:45:23 UTC+0100	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
13:44:35 UTC+0100	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Resource creation initiated
13:44:34 UTC+0100	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Requested update requires the creation of a new physical RESOURCE; hence creating one.
13:44:27 UTC+0100	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	User Initiated
13:37:55 UTC+0100	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:37:54 UTC+0100	UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:37:52 UTC+0100	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	

© 2006 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Using a template to create and manage a stack

CloudFormation Management Console

CloudFormation Management C... +

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ ianmas-aws Inland Help

Actions Filter: Active By Name: Showing 0 stacks

Stack Name	Created Time	Status	Description
------------	--------------	--------	-------------

Events +

2014-06-23	Status	Type	Logical ID	Status Reason
13:53:46 UTC+0100	DELETE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceDemoStack	
13:53:45 UTC+0100	DELETE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
13:53:17 UTC+0100	DELETE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	
13:52:55 UTC+0100	DELETE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	User Initiated
13:45:53 UTC+0100	UPDATE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceDemoStack	
13:45:53 UTC+0100	DELETE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
13:45:27 UTC+0100	DELETE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	
13:45:25 UTC+0100	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	
13:45:23 UTC+0100	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
13:44:35 UTC+0100	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Resource creation initiated
13:44:34 UTC+0100	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Requested update requires the creation of a new physical resource; hence creating one.
13:44:27 UTC+0100	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	User Initiated
13:44:26 UTC+0100	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	User Initiated

Feedback

Cleans Up Resources

Using a template to create and manage a stack

EC2 Management Console

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ Ianmas-AWS

EC2 Dashboard Events Tags Reports Limits

Instances Instances Spot Requests Reserved Instances

Images AMIs Bundle Tasks

Elastic Block Store Volumes Snapshots

Network & Security Security Groups Elastic IPs Placement Groups Load Balancers Key Pairs Network Interfaces

Auto Scaling Launch Configurations Auto Scaling Groups

Launch Instance Connect Actions

Filter: All Instances All Instance types StackName

1 to 2 of 2 Instances

Name	Instance ID	Instance Type	Availability Zone	Instance State
	i-866872bc5	m3.medium	eu-west-1a	terminated
	i-02a603f92	m3.medium	eu-west-1c	terminated

Instance: i-866872bc5 Public DNS: -

Description	Status Checks	Monitoring	Tags
Instance ID: i-866872bc5	Instance state: terminated	Public DNS: -	
	Instance type: m3.medium	Public IP: -	Elastic IP: -
	Private DNS: -	Availability zone: eu-west-1a	Security groups: -
	Private IPs: -	Scheduled events: -	
	Secondary private IPs: -		AMI ID: ami-amzn-ami-hvm-2014.09.2.x86_64-gp2 (ami-d0705ba)
	VPC ID: -		

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

Help Pack

Using a template to create and manage a stack via the AWS CLI

```
aws cloudformation create-stack  
  --stack-name ec2InstanceCmdLineDemo  
  --template-url https://s3-eu-west-1.amazonaws.com/cf-  
templates-1fhelrvrdrbr-eu-west-1/2014174d0r-ec2Instance.template  
  --parameters ParameterKey=KeyPair,ParameterValue=ManagementKeyPair
```

Returns the details of the created stack, in the output format of your choice

```
arn:aws:cloudformation:eu-west-1:554625704737:stack/ec2InstanceCmdLineDemo/  
42cc6150-fad7-11e3-8f4d-5017e1aef4e7
```

Using a template to create and manage a stack via the AWS CLI

The screenshot shows the CloudFormation Management Console interface. At the top, there are tabs for 'Create Stack', 'Update Stack', and 'Delete Stack'. Below that, a search bar with 'Filter: Active' and 'By Name' dropdown. A table lists one stack:

Stack Name	Created Time	Status	Description
ec2InstanceCmdLineDemo	2014-06-23 14:06:59 UTC+0100	CREATE_COMPLETE	Create an EC2 instance running the latest Amazon Linux AMI.

Below the table, there's a navigation bar with tabs: Overview, Outputs, Resources, Events (which is selected), Template, Parameters, Tags, and Stack Policy. The 'Events' tab displays a log of events for the stack:

Date	Status	Type	Logical ID	Status Reason
2014-06-23	CREATE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceCmdLineDemo	
14:07:54 UTC+0100	CREATE_COMPLETE	AWS::EC2::Instance	Ec2Instance	
14:07:52 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	Resource creation initiated
14:07:04 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance	
14:07:03 UTC+0100	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceCmdLineDemo	User Initiated
14:06:59 UTC+0100	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack		

At the bottom, there are links for 'Feedback', 'Privacy Policy', and 'Terms of Use'. The footer contains copyright information: © 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Other AWS CLI actions for CloudFormation

cancel-update-stack	get-stack-policy
create-stack	get-template
delete-stack	list-stack-resources
describe-stack-events	list-stacks
describe-stack-resource	set-stack-policy
describe-stack-resources	update-stack
describe-stacks	validate-template

As usual, you can get more details via the AWS CLI

```
$ aws cloudformation update-stack help
```

Help via the AWS CLI

```
$ aws cloudformation update-stack help
```

SYNOPSIS

```
update-stack
--stack-name <value>
[--template-body <value>]
[--template-url <value>]
[--use-previous-template | --no-use-previous-template]
[--stack-policy-during-update-body <value>]
[--stack-policy-during-update-url <value>]
[--parameters <value>]
[--capabilities <value>]
[--stack-policy-body <value>]
[--stack-policy-url <value>]
[--notification-arns <value>]
```

```
$ aws cloudformation update-stack help
```

SYNOPSIS

```
update-stack
--stack-name <value>
[--template-body <value>]
[--template-url <value>]
[--use-previous-template | --no-use-previous-template]
[--stack-policy-during-update-body <value>]
[--stack-policy-during-update-url <value>]
[--parameters <value>]
[--capabilities <value>]
[--stack-policy-body <value>]
[--stack-policy-url <value>]
[--notification-arns <value>]
```

WORKING WITH AWS RESOURCES

Designed to use your existing experience with AWS

Each resource has a set of parameters with names that are identical to the names used to create the resources through their native API

```
"myVolume" : {  
    "Type" : "AWS::EC2::Volume",  
    "Properties" : {  
        "Size" : "10",  
        "SnapshotId" : "snap-7b8fd361",  
        "AvailabilityZone" : "eu-west-1a"  
    }  
}
```

This example defines an Amazon EBS Volume with a logical name ‘myVolume’. Its type is “AWS::EC2::Volume”

If you’ve used EBS previously, the properties should look very familiar

```
"InstanceSecurityGroup" : {  
    "Type" : "AWS::EC2::SecurityGroup",  
    "Properties" : {  
        "GroupDescription" : "Enable SSH access via port 22",  
        "SecurityGroupIngress" : [ {  
            "IpProtocol" : "tcp",  
            "FromPort" : "22",  
            "ToPort" : "22",  
            "CidrIp" : "0.0.0.0/0"  
        } ]  
    }  
}
```

Creating a Security Group resource

Supported AWS Services:

- Auto Scaling
- Amazon CloudFront
- AWS CloudWatch
- Amazon DynamoDB
- Amazon EC2
- Amazon ElastiCache
- AWS Elastic Beanstalk
- AWS Elastic Load Balancing
- AWS Identity and Access Management
- Amazon RDS
- Amazon Redshift
- Amazon Route 53
- Amazon S3
- Amazon SimpleDB
- Amazon SNS
- Amazon SQS
- Amazon VPC

REFERENCING THE PROPERTIES OF ANOTHER RESOURCE

```
{ "Resources" : {  
    "Ec2Instance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],  
            "KeyName" : "mykey",  
            "ImageId" : "ami-7a11e213"  
        }  
    },  
    "InstanceSecurityGroup" : {  
        "Type" : "AWS::EC2::SecurityGroup",  
        "Properties" : {  
            "GroupDescription" : "Enable SSH access via port 22",  
            "SecurityGroupIngress" : [ {  
                "IpProtocol" : "tcp",  
                "FromPort" : "22",  
                "ToPort" : "22",  
                "CidrIp" : "0.0.0.0/0" } ]  
        }  
    }  
}
```

```
{ "Resources" : {  
    "Ec2Instance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" }, ,  
"MyExistingSG" ], "KeyName" : "mykey",  
            "ImageId" : "ami-7a11e213" }  
    },  
    "InstanceSecurityGroup" : {  
        "Type" : "AWS::EC2::SecurityGroup",  
        "Properties" : {  
            "GroupDescription" : "Enable SSH access via port 22",  
            "SecurityGroupIngress" : [ {  
                "IpProtocol" : "tcp",  
                "FromPort" : "22",  
                "ToPort" : "22",  
                "CidrIp" : "0.0.0.0/0" } ]  
        }  
    }  
}
```

REFERENCING INPUT PARAMETERS

Input Parameters

```
{  
  "Parameters" : {  
    "KeyPair" : {  
      "Description" : "The EC2 Key Pair to allow SSH access to the instance",  
      "Type" : "String"  
    },  
    "Resources" : {  
      "Ec2Instance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
          "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],  
          "KeyName" : { "Ref" : "KeyPair" },  
          "ImageId" : ""  
        },  
        ...  
      } } }
```

Input
Parameters

Create A New Stack

Services VPC EC2 S3 RDS CloudFront Edit Ianmas @ Ianmas-aws Ireland Help

Select Template

Specify Parameters

Options Review

Specify values or use the default values for the parameters that are associated with your AWS CloudFormation template.

Parameters

KeyPair ManagementKeyPair The EC2 Key Pair to allow SSH access to the instance

Cancel Previous Next

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

```
"WordPressUser": {  
    "Default": "admin",  
    "Description" : "The WordPress database admin account username",  
    "Type": "String",  
    "MinLength": "1",  
    "MaxLength": "16",  
    "AllowedPattern" : "[a-zA-Z][a-zA-Z0-9]*"  
},
```

Validate your input parameters with :

Maxlength, MinLength, MaxValue, MinValue, AllowedPattern, AllowedValues

CONDITIONAL VALUES

```
{"Mappings" : {  
    "RegionMap" : {  
        "us-east-1" : { "AMI" : "ami-76f0061f" },  
        "us-west-1" : { "AMI" : "ami-655a0a20" },  
        "eu-west-1" : { "AMI" : "ami-7fd4e10b" },  
        "ap-southeast-1" : { "AMI" : "ami-72621c20" },  
        "ap-northeast-1" : { "AMI" : "ami-8e08a38f" } } },  
"Resources" : {  
    "Ec2Instance" : {  
        "Type" : "AWS::EC2::Instance",  
        "Properties" : {  
            "KeyName" : { "Ref" : "KeyName" },  
            "ImageId" : {  
                "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ]  
            }  
        }  
    } } }
```

Other intrinsic functions and pseudo parameters

Intrinsic functions

Fn::Base64

Fn::FindInMap

Fn::GetAtt

Fn::GetAZs

Fn::Join

Fn::Select

Ref

Pseudo parameters

AWS::NotificationARNs

AWS::Region

AWS::StackId

AWS::StackName

Working with non-AWS Resources

Defining custom resources allows you
to include non-AWS resources in a
CloudFormation stack

More on Custom Resources in ‘AWS CloudFormation under the Hood’ from re:Invent 2013: <http://youtu.be/ZhGMaw67Yu0>
AWS CloudFormation Custom Resource Walkthrough documentation:

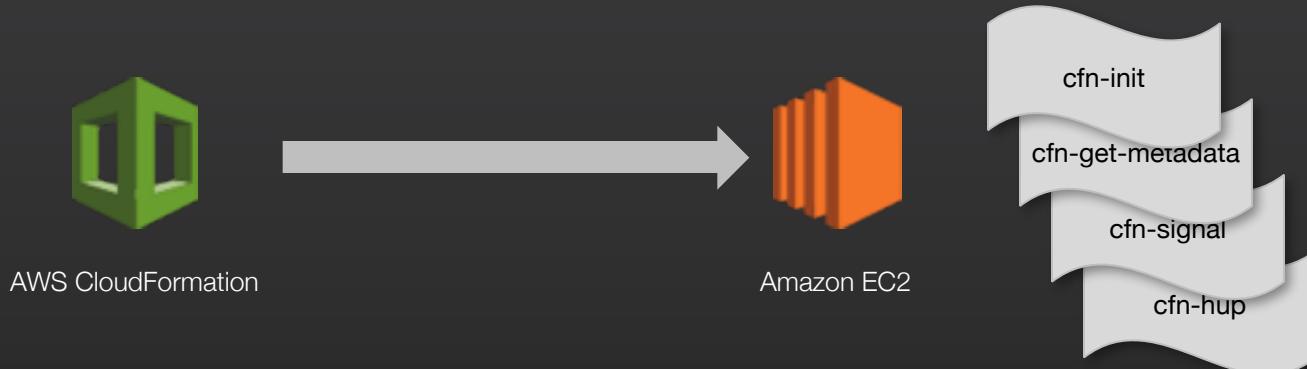
docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/crpg-walkthrough.html

BOOTSTRAPPING APPLICATIONS AND HANDLING UPDATES

Option 1: Continue to use EC2 UserData, which is available as a property of AWS::EC2::Instance resources

```
"Resources" : {
    "Ec2Instance" : {
        "Type" : "AWS::EC2::Instance",
        "Properties" : {
            "KeyName" : { "Ref" : "KeyName" },
            "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
            "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" : "AWS::Region" }, "AMI" ] },
            "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
                "#!/bin/bash -ex",
                "yum -y install gcc-c++ make",
                "yum -y install mysql-devel sqlite-devel",
                "yum -y install ruby-rdoc rubygems ruby-mysql ruby-devel",
                "gem install --no-ri --no-rdoc rails",
                "gem install --no-ri --no-rdoc mysql",
                "gem install --no-ri --no-rdoc sqlite3",
                "rails new myapp",
                "cd myapp",
                "rails server -d",
                "curl -X PUT -H 'Content-Type: application/json' --data-binary '{\"Status\": \"SUCCESS\",
                \"Reason\": \"The application myapp is ready\",
                \"UniqueId\": \"myapp\",
                \"Data\": \"Done\"}' \"\",
                \"\", {"Ref": "WaitForInstanceWaitHandle"}, \"\n\" ]]} }
        }
    }
}
```

Option 2: AWS CloudFormation provides helper scripts for deployment within your EC2 instances



Metadata Key – **AWS::CloudFormation::Init**

Cfn-init reads this metadata key and installs the packages listed in this key (e.g., httpd, mysql, and php). Cfn-init also retrieves and expands files listed as sources.

Installing Packages & Expanding Files

```
"Resources" : {
    "WebServer": {
        "Type": "AWS::EC2::Instance",
        "Metadata" : {
            "Comment1" : "Configure the bootstrap helpers to install the Apache Web Server and PHP",
            "Comment2" : "The website content is downloaded from the CloudFormationPHPSample.zip file",
        }
        "AWS::CloudFormation::Init" : {
            "config" : {
                "packages" : {
                    "yum" : {
                        "mysql"      : [],
                        "mysql-server" : [],
                        "mysql-libs"   : [],
                        "httpd"       : [],
                        "php"         : [],
                        "php-mysql"   : []
                    }
                },
                "sources" : {
                    "/var/www/html" : "https://s3.amazonaws.com/cloudformation-examples/CloudFormationPHPSample.zip"
                }
            }
        }
    }
}
```

The `UserData` key allows you to execute shell commands.

This template issues two shell commands: the first command installs the AWS CloudFormation helper scripts; the second executes the `cfn-init` script.

```
"Properties": {
    "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
                                      { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] },
    "InstanceType" : { "Ref" : "InstanceType" },
    "SecurityGroups" : [ { "Ref" : "WebServerSecurityGroup" } ],
    "KeyName" : { "Ref" : "KeyName" },
    "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
        "#!/bin/bash -v\n",
        "yum update -y aws-cfn-bootstrap\n",

        "# Install packages\n",
        "/opt/aws/bin/cfn-init -s ", { "Ref" : "AWS::StackName" }, " -r WebServer ",
        " --region ", { "Ref" : "AWS::Region" }, " || error_exit 'Failed to run cfn-init'\n"
    ]]} }
},
```

The files key allows you to write files to the instance filesystem

```
"files" : {  
    "/tmp/setup.mysql" : {  
        "content" : { "Fn::Join" : [ "", [  
            "CREATE DATABASE ", { "Ref" : "DBName" }, ";\\n",  
            "GRANT ALL ON ", { "Ref" : "DBName" }, ".* TO '", { "Ref" : "DBUsername" }, "'@localhost IDENTIFIED BY  
'", { "Ref" : "DBPassword" }, "';\\n"  
        ] ]},  
        "mode" : "000644",  
        "owner" : "root",  
        "group" : "root"  
    }  
}
```

The services key allows you ensures that the services are not only running when cfn-init finishes (ensureRunning is set to true); but that they are also restarted upon reboot (enabled is set to true).

```
"services" : {  
    "sysvinit" : {  
        "mysqld" : {  
            "enabled"      : "true",  
            "ensureRunning" : "true"  
        },  
        "httpd" : {  
            "enabled"      : "true",  
            "ensureRunning" : "true"  
        }  
    }  
}
```

Yes!

All this functionality is available for
Windows instances too!

What about Chef?

and/or

What about Puppet?

AWS CloudFormation Articles and Tutorials

Create Free Account

AWS CloudFormation gives developers and systems administrators an easy way to create a collection of related AWS resources and provision them in an orderly and predictable fashion. The following articles and documents provide guidance on building templates and using the various AWS CloudFormation features to provision your AWS resources.

Bootstrapping Applications via AWS CloudFormation

AWS CloudFormation gives you an easy way to create the set of resources such as Amazon EC2 instance, Amazon RDS database instances and Elastic Load Balancers needed to run your application. The template describes what resources you need and AWS CloudFormation takes care of how: provisioning the resources in an orderly and predictable fashion, handling and recovering from any failures or issues. While AWS CloudFormation takes care of provisioning all the resources, it raises the obvious question of how your application software is deployed, configured and executed on the Amazon EC2 instances. There are many options, each of which has implications on how quickly your application is ready and how flexible you need to be in terms of deploying new versions of the software.

[Read on... !\[\]\(8c4731260733f7611bbd83a86c6983ee_img.jpg\)](#)

Integrating AWS CloudFormation with Opscode Chef

AWS CloudFormation can help you to configure and/or install your application as well as how to bootstrap deployment and management tools that you may already use in your environment. Chef is an open source infrastructure automation solution from Opscode, written in Ruby, that allows you to automate the configuration of your systems and the applications that sit on top of it. AWS CloudFormation and Chef can be used together to automate your entire deployment and management processes, from your AWS resources through to your application artifacts.

[Read on... !\[\]\(2cc714b41df9e90bf80af98ed602c9fc_img.jpg\)](#)

Integrating AWS CloudFormation with Puppet

AWS CloudFormation can help you to configure and/or install your application as well as how to bootstrap deployment and management tools that you may already use in your environment. Puppet is an open source platform for provisioning, configuring and patching applications and operating system components. AWS CloudFormation and Puppet can be used together to automate your entire deployment and management processes, from your AWS resources through to your application artifacts.

[Read on... !\[\]\(9a939841023aa50a20146f1579be1107_img.jpg\)](#)

SUMMARY

- ① An easy way to create & manage a collection of AWS resources
- ② Allows orderly and predictable provisioning and updating of resources
- ③ Allows you to version control your AWS infrastructure
- ④ Deploy and update stacks using console, command line or API

**RESOURCES YOU CAN USE
TO LEARN MORE**

aws.amazon.com/cloudformation/

Getting Started with AWS CloudFormation:

aws.amazon.com/cloudformation/getting-started/

AWS CloudFormation Templates & Samples:

aws.amazon.com/cloudformation/aws-cloudformation-templates/

AWS cfncluster HPC deployment framework:

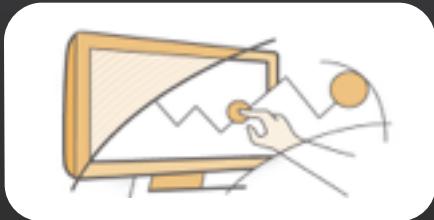
github.com/awslabs/cfncluster/

CloudFormation Template to Deploy Wordpress

https://s3-us-west-1.amazonaws.com/cloudformation-templates-us-west-1/WordPress_Multi_AZ.template

AWS Training & Certification

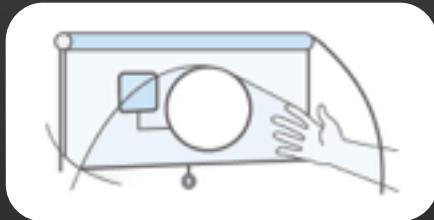
Self-Paced Labs



Try products, gain new skills, and get hands-on practice working with AWS technologies

[aws.amazon.com/training/
self-paced-labs](https://aws.amazon.com/training/self-paced-labs)

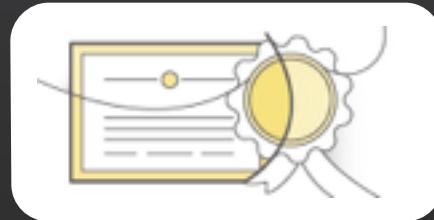
Training



Build technical expertise to design and operate scalable, efficient applications on AWS

aws.amazon.com/training

Certification



Validate your proven skills and expertise with the AWS platform

aws.amazon.com/certification



15 APRIL 2015

aws.amazon.com/summits/london/

AWS CloudFormation will be featured in the Deep Dive: Infrastructure-as-Code breakout session



Follow us for more
events & webinars



Ian Massingham — Technical Evangelist

 @ianMmmm



@AWS_UK for local AWS events & news



@AWScloud for Global AWS News & Announcements