



Masterclass

AWS CloudFormation



Ian Massingham – Technical Evangelist

ianmas@amazon.com

[@IanMmm](https://twitter.com/IanMmm)



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

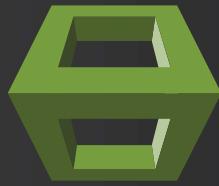
- An easy way to create & manage a collection of AWS resources
- Allows orderly and predictable provisioning and updating of resources
- Allows you to version control your AWS infrastructure
- Deploy and update stacks using console, command line or API
- You only pay for the resources you create



Transparent and Open

Don't reinvent the wheel

Declarative & Flexible



AWS CloudFormation

No Extra Charge

Customized via Parameters

Integration Ready

CloudFormation - Components & Technology

Template



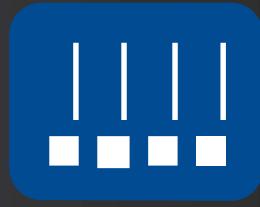
JSON formatted file

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

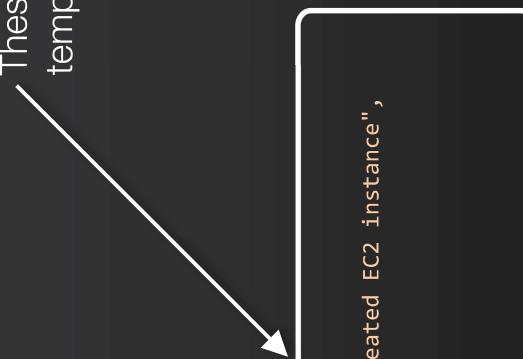
```
{  
  "Description": "Create an EC2 instance running the latest Amazon Linux AMI.",  
  "Parameters": {  
    "KeyPair": {  
      "Description": "The EC2 Key Pair to allow SSH access to the instance",  
      "Type": "String"  
    },  
    "Resources": {  
      "EC2Instance": {  
        "Properties": {  
          "ImageId": "ami-9d23aeea",  
          "InstanceId": "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"  
        }  
      },  
      "AMSTemplateFormatVersion": "2010-09-09"  
    }  
}
```

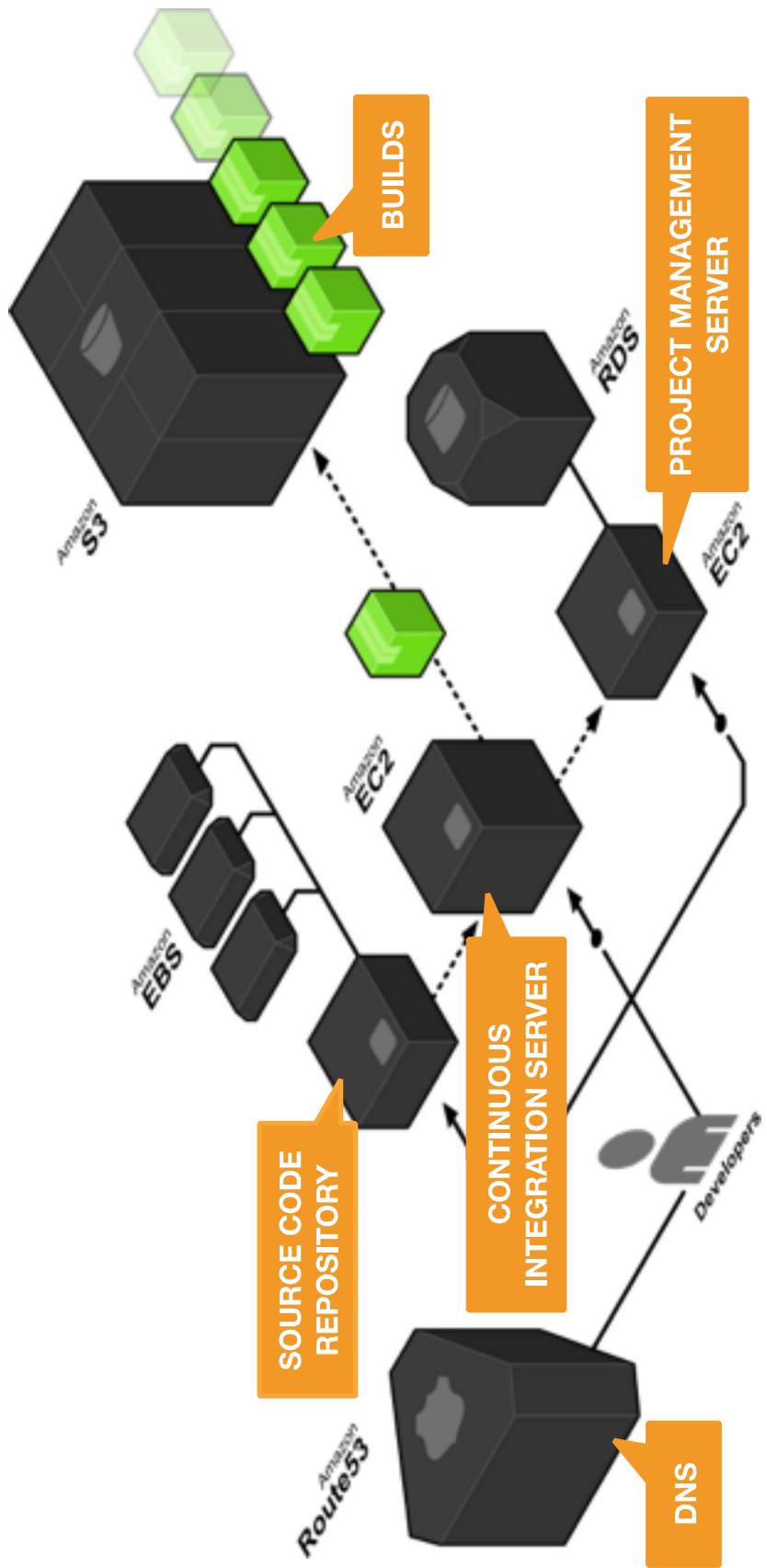
Includes statically defined properties
(ImageID & Instance Type) plus a reference to
the KeyPair parameter. ImageID is the AMI
specific to the region that you will launch this
stack in, in this case the eu-west-1 region

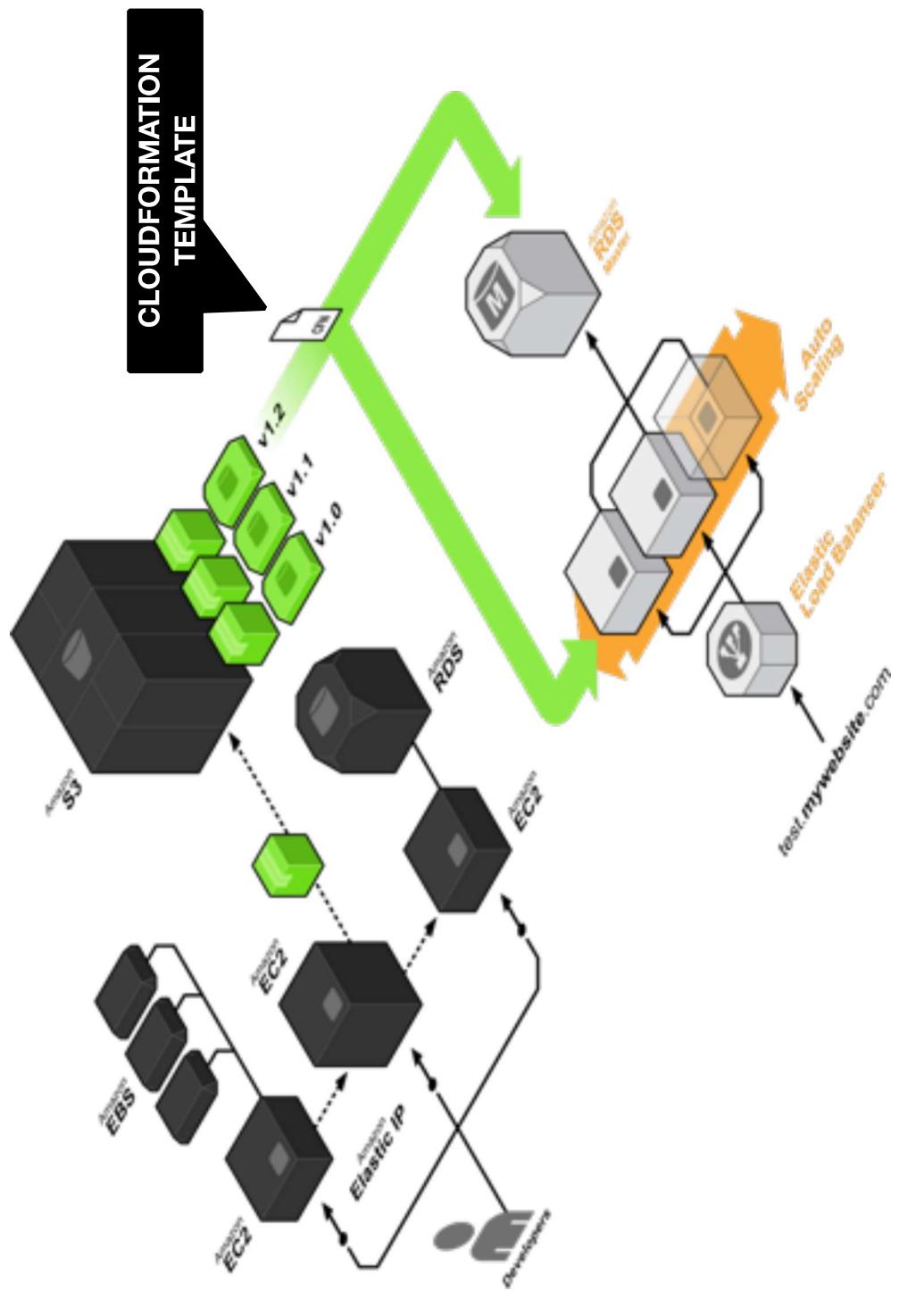
A Simple Template that creates an EC2 Instance

```
{  
  "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"  
      }  
    },  
    "AMSTemplateFormatVersion": "2010-09-09"  
  }  
}
```

These outputs will be returned once the template has completed execution







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

Installation Instructions

卷之三

Example

<http://www.elsevier.com/locate/jat> <http://www.elsevier.com/locate/jat>

CREATING & MANAGING STACKS

Using a template
to create and
manage a stack

Using a template to create and manage a stack

The screenshot shows the AWS Management Console with the CloudFormation service selected. A specific stack named 'CloudFront' is highlighted with a red box. The stack details page displays the following resources:

- CloudFront Distribution
- Lambda@Edge Function
- CloudWatch Metrics



AWS Marketplace
Find & Buy software,
Launch with 1 Click and
Pay by the hour.

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Feedback

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Feedback

See Us In Person
Join Your Peers

Feedback

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation Management Console. At the top, there are navigation links for CloudFormation Management, VPC, Services, Actions, Filter: Active, By Name:, and a search bar. Below the navigation is a toolbar with icons for Create, Delete, Edit, and Help.

The main content area has two sections:

- Create a Stack:** This section explains what CloudFormation is and provides a link to "Create New Stack". A blue button labeled "Create New Stack" is highlighted with an orange rectangle.
- Create a Template from your Existing Resources:** This section explains how to capture existing resources and provides a link to "Launch CloudFormation".

At the bottom of the page, there are footer links for Feedback, Terms of Use, Privacy Policy, and a copyright notice. There are also links for Help, Support, and Contact Us.

Using a template to create and manage a stack

The screenshot shows the 'Create A New Stack' wizard in the AWS CloudFormation console. The steps are:

- Create A New Stack** (highlighted)
- Services** (dropdown menu)
- VPC**
- EC2**
- RDS**
- CloudFront**
- Logs**
- Help**

Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

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 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 No file selected

Specify an Amazon S3 template URL

Feedback

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

Using a template to create and manage a stack

The screenshot shows the 'Create A New Stack' wizard in the AWS CloudFormation console. The current step is 'Select Template'. The navigation bar at the top includes 'Create A New Stack', 'Services' (with VPC selected), 'EC2', 'SQS', 'RDS', 'CloudFront', 'Edit', 'Inbound', and 'Help'.

Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

Stack

An AWS CloudFormation stack is a collection of nested 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 + 1

Upload a template to Amazon S3 Browse No file selected.

Specify an Amazon S3 template URL

Feedback

Cancel Next

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

Using a template to create and manage a stack

The screenshot shows the 'Create A New Stack' wizard in the AWS CloudFormation console. The current step is 'Select Template'. The navigation bar at the top includes 'Create A New Stack', 'Services' (selected), 'VPC', 'EC2', 'SQS', 'RDS', 'CloudFront', 'Edit', 'Inbound', 'Outbound', and 'Help'.

Select Template

Specify a stack name and then select the template that describes the stack that you want to create.

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...

Specify an Amazon S3 template URL

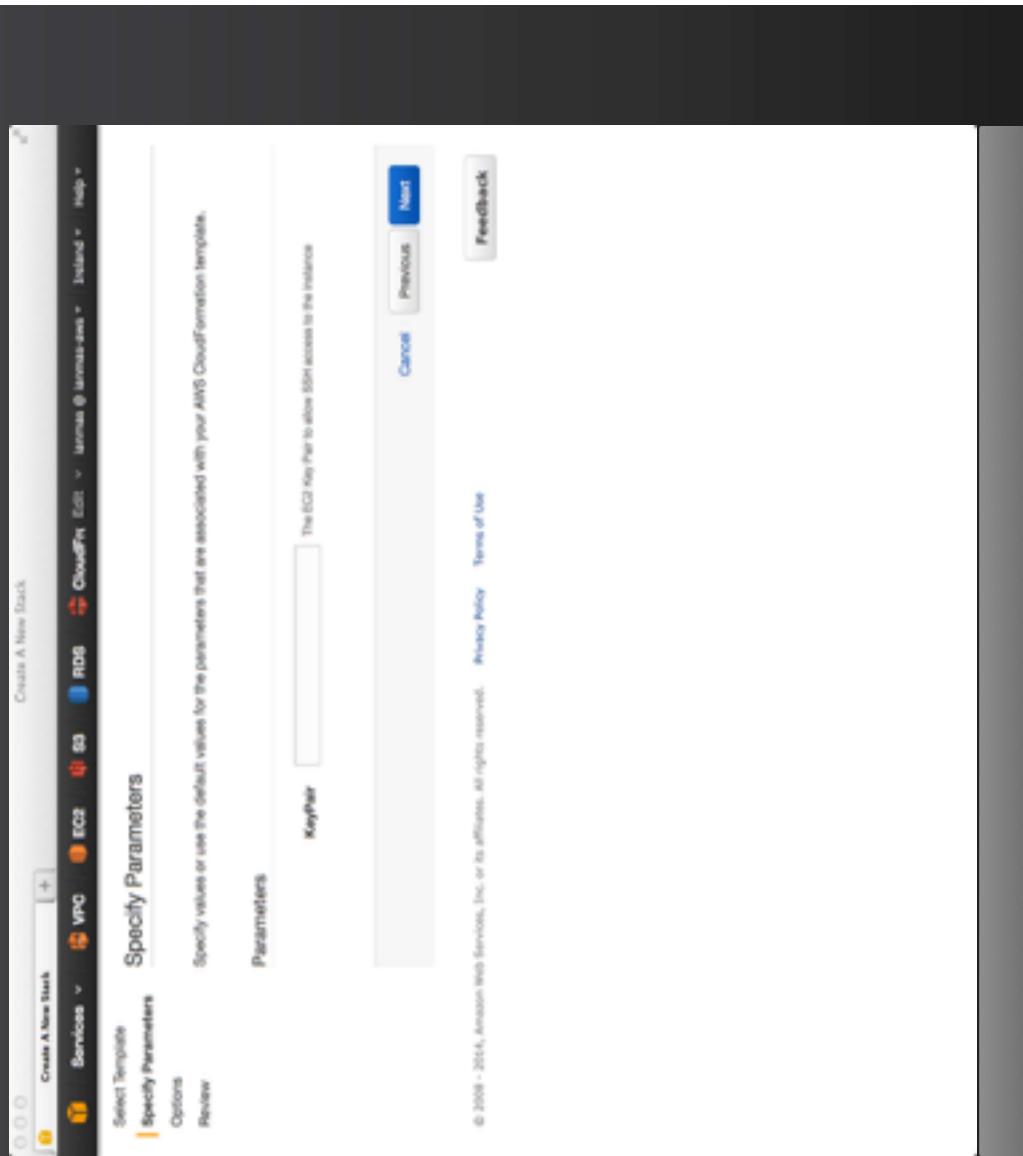
Next Step

Cancel

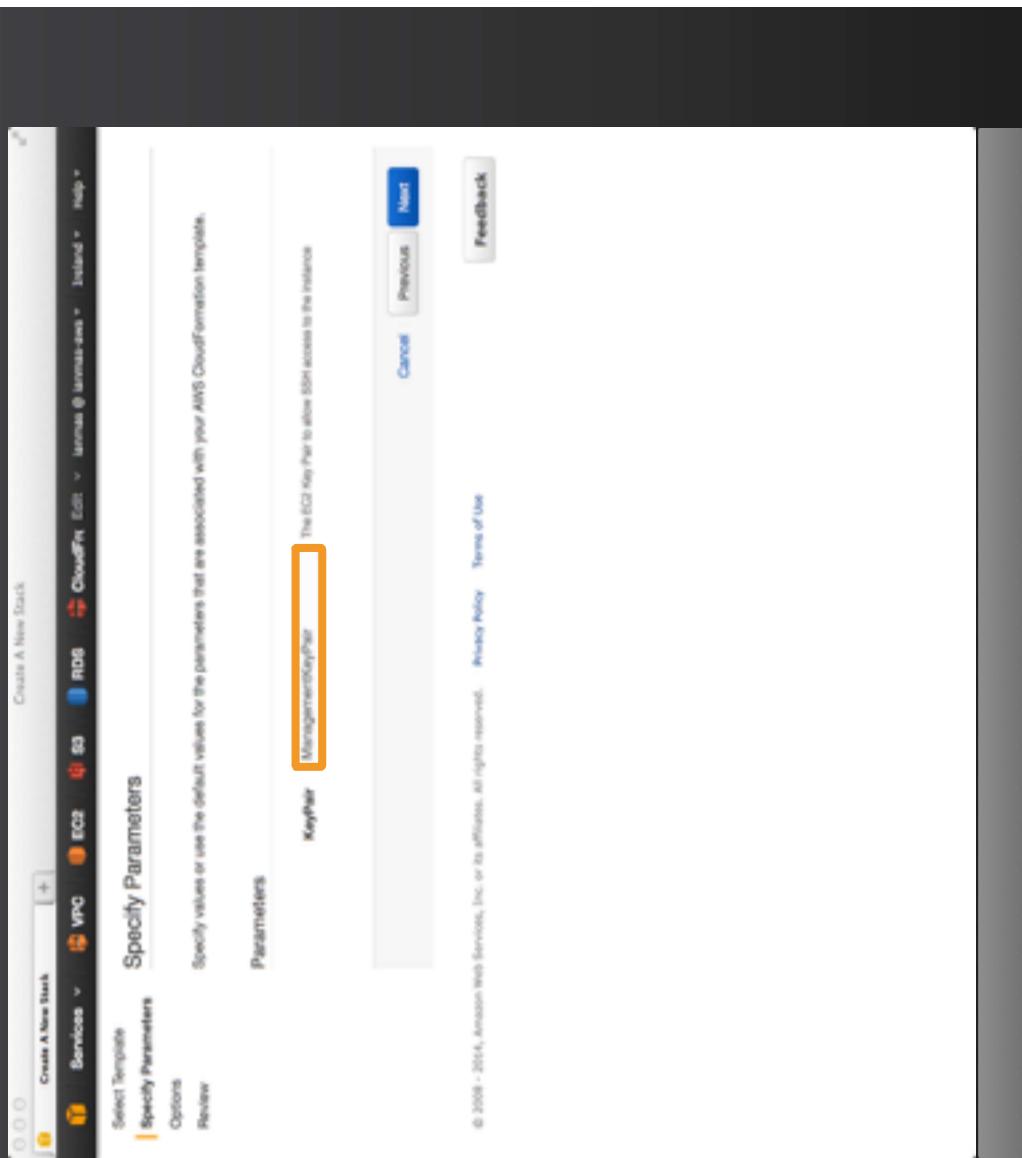
Feedback

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

Using a template to create and manage a stack



Using a template to create and manage a stack



Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation 'Create A New Stack' wizard at the 'Options' step. The top navigation bar includes 'Create A New Stack', 'Services', 'VPC', 'EC2', 'SQS', 'RDS', 'CloudFront', 'Edit', 'Lambda', 'Ingestion', and 'Help'. The left sidebar has tabs for 'Select Template', 'Specify Parameters', 'Options' (which is selected), 'Tags', and 'Review'. The main area shows a table for adding tags:

Key	Value
name	StackName

A blue '+' button is at the bottom right of the table. Below the table, there's an 'Advanced' section with a note: 'You can set additional options for your stack, like notification options and a stack policy. Learn more.' At the bottom right of the wizard are 'Cancel', 'Previous', and 'Next' buttons.

At the very bottom of the page, there's a footer with links: '© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation console during the creation of a new stack. The top navigation bar includes links for 'Create A New Stack', 'Services', 'VPC', 'EC2', 'SQS', 'RDS', 'CloudFront', 'Edit', 'Lambda @ Lambda@Edge', 'Inbound', and 'Help'. The main area is titled 'Review' and contains several tabs: 'Select Template', 'Specify Parameters', 'Options', 'Review', and 'Template'. The 'Template' tab is selected, showing a template named 'ec2instancedDemostack' with a URL of 'https://s3.amazonaws.com/ctf-templates/1/tempyardfor-eu-west-1/029-41740-jqy'. Below this, the 'Description' field states 'Create an EC2 instance running the newest Amazon Linux AMI.', and the 'Estimate cost' field shows 'Link is not available'. A 'Parameters' section lists 'KeyPair' (ManagementKeyPair) and 'Create IAM resources' (False). An 'Options' section includes 'Tags' (Name: StackName) and 'Advanced' settings for 'Notification' (Timeout: none, Rollback on Failure: Yes). At the bottom right, there are 'Cancel', 'Previous', and a prominent blue 'Create' button, which is highlighted with a yellow border.

Using a template to create and manage a stack

The screenshot shows two pages from the CloudFormation Management Console:

Stack Management (Left):

- Header: CloudFormation Management Console
- Actions: Actions, Create, Delete, Refresh
- Filter: Active, By Name:
- Table columns: Stack Name, Created Time, Status, Description
- Data row: ec2instanceDemoStack, 2014-06-23 12:59:47 UTC+0100, CREATE_COMPLETE, Create an EC2 instance running the newest AmiName

Events (Right):

- Header: Showing 1 stack
- Actions: Actions, Create, Delete, Refresh
- Table columns: Status, Type, Logical ID, Status, Reason
- Data row: 2014-06-23 12:59:47 UTC+0100, CREATE_IN_PROGRESS, ec2instanceDemoStack, User Initiated
- Note: 4 more events available to display

Using a template to create and manage a stack

The screenshot shows the CloudFormation Management Console interface. At the top, there are service icons for VPC, EC2, RDS, CloudFront, CloudWatch Metrics, Lambda, and CloudWatch Logs. Below the header, there are tabs for CloudFormation Management, Services, and Actions.

The main area displays a table of stacks:

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

Below the table, a section titled "Events" shows log entries:

Date	Status	Type	Logical ID	Status	Reason
2014-06-23					
13:00:43 UTC+0100	CREATE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceDemoStack		
13:00:43 UTC+0100	CREATE_COMPLETE	AWS::EC2::Instance	Ec2Instance		Resource creation initiated
12:59:54 UTC+0100	CREATE_IN_PROGRESS	AWS::EC2::Instance	AWSSc1		
12:59:52 UTC+0100	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	Ec2InstanceDemoStack		
12:59:47 UTC+0100	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack			User initiated

At the bottom of the page, there are links for Feedback and Feedback, and copyright information:

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.
AWS CloudFormation™ is a trademark of Amazon.com, Inc., or its affiliates.

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation Management Console interface. At the top, there are service icons for VPC, EC2, RDS, CloudFront, CloudWatch Metrics, Lambda, and CloudWatch Logs. Below the header, there are tabs for 'CloudFormation Management C...', 'Services', and 'CloudFormation'. A search bar is present above a navigation menu with links for 'Actions', 'Filter: Active', 'By Name...', 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', and 'Stack Policy'. The main content area displays a table with one row for the stack 'ec2InstanceDemoStack'. The table columns are 'Stack Name', 'Created Time', 'Status', and 'Description'. The status is 'CREATE_COMPLETE'. The description is 'Create an EC2 instance running the newest Amazon Linux AMI'. The bottom right corner of the screenshot contains footer text: '© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use'.

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

Using a template to create and manage a stack

The image consists of two screenshots from the AWS CloudFormation Management Console.

The top screenshot shows a list of stacks. One stack, named "ec2InstanceDemoStack", is listed with a status of "CREATE_COMPLETE". The stack was created on 2014-06-23 12:59:47 UTC+0100. A description for the stack states: "Create an EC2 instance running the newest Amazon Linux AMI." The "Outputs" section shows one output:

Key	Value	Description
InstanceId	i-02a60702	The InstanceId of the newly created EC2 instance

The bottom screenshot shows the AWS Lambda function configuration page. It displays the same output variable "InstanceId" with the value "i-02a60702".

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

[Feedback](#)

[Feedback](#)

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation console with a stack named "ec2InstanceTemplate". The status is "CREATE FAILED" with the reason "Template error: /tmp/stack/ec2InstanceTemplate/stack-1443338588838885585/CloudFormationStackTemplate.json:16:1: Invalid character '}' at end of JSON object". The stack creation started at 2018-09-09T10:25:46+00:00 and last updated at 2018-09-09T10:25:46+00:00.

```
1 {
  "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-0d929000",
        "InstanceIdType": "t3.medium",
        "KeyName": {
          "Ref": "KeyPair"
        },
        "SecurityGroups": "ssh-tools"
      },
      "Type": "AWS::EC2::Instance"
    }
  },
  "Outputs": {
    "InstanceId": {
      "Description": "The InstanceId of the newly created EC2 instance",
      "Value": {
        "Ref": "EC2Instance"
      }
    }
  },
  "Metadata": {
    "AWS::CloudFormation::Interface": {
      "ParameterOrder": [
        "KeyPair"
      ]
    }
  }
}
```

Incorrect Syntax

Using a template to create and manage a stack

The screenshot shows the CloudFormation Management Console interface. At the top, there are navigation links for CloudFormation Management, Services (VPC, EC2, RDS), and Help. Below the header, there are tabs for Actions, Outputs, and Resources.

Actions tab:

- Create Stack
- Update Stack** (highlighted with an orange box)
- Delete Stack

Outputs tab:

Key	Value	Description
InstanceId	i-02a601702	The InstanceId of the newly created EC2 instance

Resources tab:

Name	Status	Description
ec2-test-stack	CREATE_COMPLETE	Create an EC2 instance running the relevant AmiId

At the bottom of the page, there are footer links for Feedback, Support, and Terms of Use.

Using a template to create and manage a stack

The screenshot shows the 'Select Template' step of a CloudFormation stack update process. At the top, there are tabs for 'Select Template', 'Specify Parameters', 'Options', and 'Review'. The 'Select Template' tab is active. Below it, there's a note about what a template is and how to update an existing stack. A dropdown menu titled 'Source' is open, showing four options:

- Upload a template in an Amazon S3 bucket.
- Browse for CloudFormation template.
- Specify an Amazon S3 template URL.
- Use existing template.

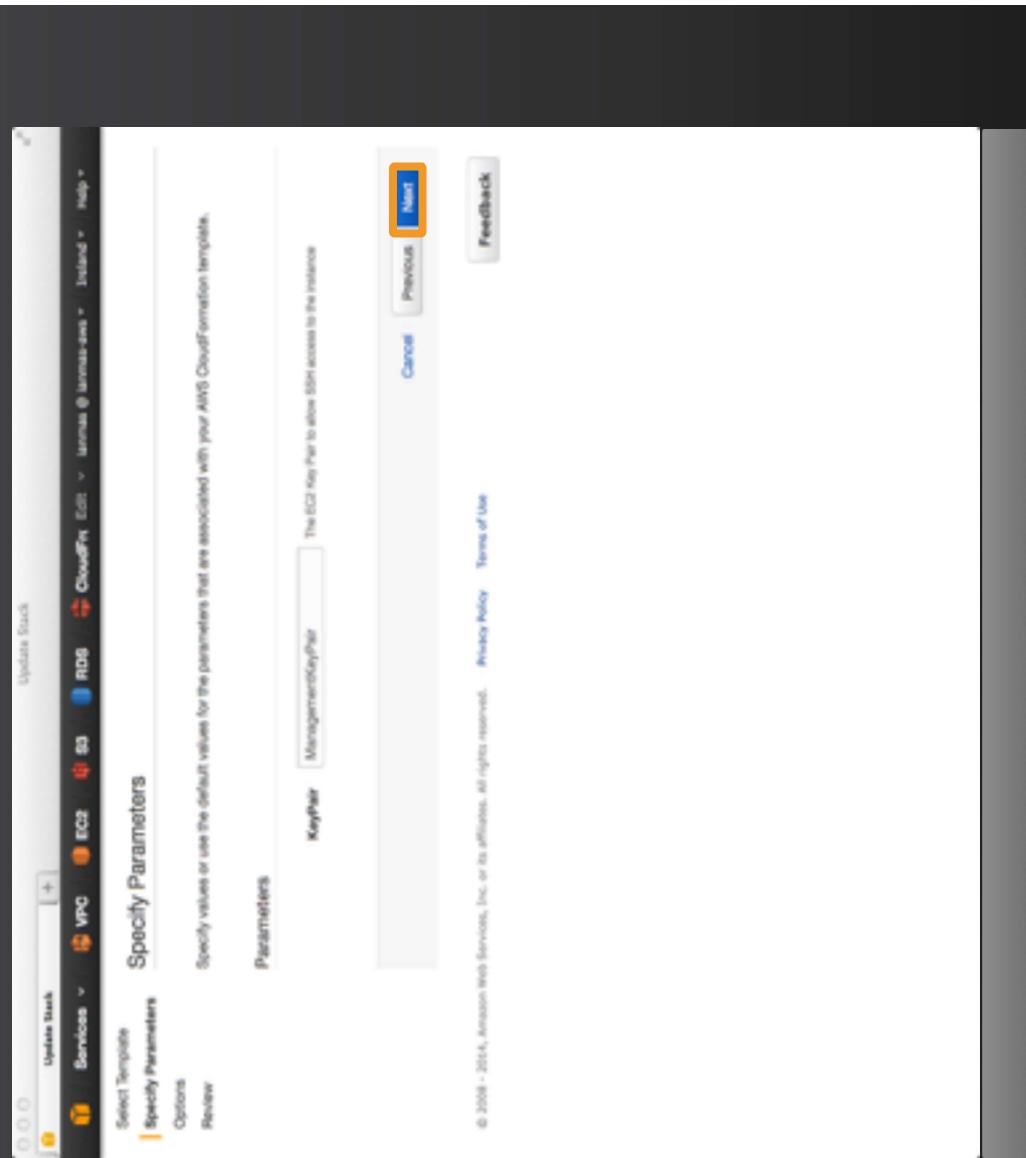
The 'Browse for CloudFormation template' option is highlighted with an orange rectangle. The 'Name' field contains 'ec2instanceDemoStack'. At the bottom right of the dialog are 'Next Step' and 'Cancel' buttons.

© 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



Using a template to create and manage a stack

The screenshot shows the CloudFormation Management Console interface. On the left, there's a navigation bar with icons for CloudWatch Metrics, CloudWatch Logs, CloudFormation Management, VPC, Services, EC2, RDS, CloudFront, and Lambda. The main area has two tabs: 'Stack Overview' and 'Events'. The 'Stack Overview' tab shows a single stack named 'ec2instanceDemoStack' with a status of 'CREATE_COMPLETE'. The 'Events' tab lists several events:

Event	Timestamp	Action	Resource	Description
13:37:55 UTC+01:00	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:37:54 UTC+01:00	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:37:52 UTC+01:00	UPDATE_COMPLETE	AWS::EC2::Instance	ec2instanceDemoStack	The following resource(s) failed to update: [Ec2Instance]. Value of SecurityGroup must be of type List of String
13:37:27 UTC+01:00	UPDATE_ROLLBACK_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack	
13:37:26 UTC+01:00	UPDATE_FAILED	AWS::EC2::Instance	ec2instance	

At the bottom, there are links for 'Feedback' and 'Feedback'.

Provides Feedback

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation Management Console interface. At the top, there's a navigation bar with links for CloudFormation Management, Services (VPC, EC2, RDS, CloudFront), and a search bar. Below the navigation, a blue banner with white text reads "Provides Feedback". The main area displays a table of stacks:

Stack Name	Created Time	Status	Description
ec2instanceDemoStack	2014-06-23 12:59:48 UTC+01:00	UPDATE_ROLLBACK_COMPLETE	Create an EC2 instance running the newest Amazone

Below the table, a section titled "Events" lists the following log entries:

- 13:37:55 UTC+01:00 UPDATE_ROLLBACK_COMPLETE AWS::CloudFormation::Stack EC2instanceDemoStack
- 13:37:54 UTC+01:00 UPDATE_ROLLBACK_COMPLETE AWS::CloudFormation::Stack EC2instanceDemoStack
- 13:37:52 UTC+01:00 UPDATE_COMPLETE AWS::EC2::Instance EC2instanceDemoStack
- 13:37:27 UTC+01:00 UPDATE_ROLLBACK_IN_PROGRESS AWS::CloudFormation::Stack EC2instanceDemoStack
- 13:37:26 UTC+01:00 UPDATE_FAILED AWS::EC2::Instance EC2instance

A message box on the right side of the screen contains the following text, which is highlighted with an orange rectangle:

Value of property SecurityGroups must be of type List of String

At the bottom of the page, there are links for "Feedback", "Feedback", and "Feedback". The footer contains copyright information and links for "Privacy Policy", "Terms of Use", "Help", "Support", "Feedback", and "Feedback".

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-0d925000",
13       "InstanceType": "t3.medium",
14       "KeyName": {
15         "Ref": "KeyPair"
16       },
17       "SecurityGroups": [
18         "ssh-tools"
19       ],
20       "Type": "AWS::EC2::Instance"
21     }
22   },
23   "Outputs": {
24     "InstanceId": {
25       "Description": "The InstanceId of the newly created EC2 instance",
26       "Value": {
27         "Ref": "EC2Instance"
28       }
29     },
30     "AMSTemplateFormatVersion": "2010-09-09"
31   }
32 }
```

Correct Syntax

Using a template to create and manage a stack

The screenshot shows the CloudFormation Management Console interface. At the top, there are service icons for CloudFormation, Management C..., VPC, EC2, RDS, and CloudFront. Below the header, a navigation bar includes 'Actions', 'Filter: Active', 'By Name:', 'Stack Name', 'Created Time', 'Status', and 'Description'. The status column shows 'UPDATE_COMPLETE' for the stack named 'ec2instanceDemoStack'. The description field contains the note 'Create an EC2 instance running the latest Amazon 1'. On the left, a sidebar titled 'Events' lists several events with their status, type, logical ID, and reason:

Status	Type	Logical ID	Reason
2014-06-23 13:45:53 UTC+01:00	UPDATE_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack
13:45:53 UTC+01:00	DELETE_IN_PROGRESS	AWS::EC2::Instance	User Initiated
13:45:27 UTC+01:00	DELETE_IN_PROGRESS	AWS::EC2::Instance	User Initiated
2014-06-23 13:45:25 UTC+01:00	UPDATE_COMPLETE, CANCELLED	AWS::CloudFormation::Stack	ec2instanceDemoStack
P_IN_PROGRESS	AWS::EC2::Instance	AWS::EC2::Instance	Ec2Instance
13:45:23 UTC+01:00	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance
13:44:56 UTC+01:00	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance
13:44:34 UTC+01:00	UPDATE_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance
13:44:27 UTC+01:00	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2instanceDemoStack
13:39:55 UTC+01:00	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack
TE	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2instanceDemoStack
13:37:54 UTC+01:00	TE_CLEANUP_IN_PROGRESS	AWS::EC2::Instance	Ec2Instance
13:37:50 UTC+01:00	UPDATE_COMPLETE	AWS::EC2::Instance	Ec2Instance

At the bottom, there is a 'Feedback' button and a footer with copyright information.

Using a template to create and manage a stack

The screenshot shows the AWS EC2 Management Console with the 'Launch Instance' wizard open. The 'Description' tab is selected, displaying instance details such as Public DNS, Instance ID, and Security Groups. A yellow box highlights the 'Description' tab. A blue box highlights the 'Security Groups' section where a security group named 'Web' is selected.

Instance	Public DNS	Instance ID	Description
i-06072ec5	ec2-54-178-124-9.eu-west-1.compute.amazonaws.com	i-06072ec5	Public DNS: ec2-54-178-124-9.eu-west-1.compute.amazonaws.com

Feedback

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use See To Smart Policy Version: November 2014 Last updated: July 14, 2014 © 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use See To Smart Policy Version: November 2014 Last updated: July 14, 2014

Replaces Resources

Using a template to create and manage a stack

The screenshot shows the CloudFormation Management Console interface. At the top, there are navigation links for Home, Services, VPC, EC2, RDS, CloudFront, and Help. Below the navigation bar, there's a search bar and a 'Create Stack' button.

The main area displays a table for the new stack:

Name:	Status	Description
ec2demostack	CREATE_IN_PROGRESS	Create an EC2 instance running the latest Amazon Linux AMI

Below the table, there's a section titled 'Events' showing the stack's history:

Time	Status	Type	Logical ID	Status	Reason
2014-06-23 13:45:53 UTC+01:00	UPDATE_COMPLETE	AWS::CloudFormation::Stack	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:45:53 UTC+01:00	DELETE_IN_PROGRESS	AWS::EC2::Instance	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:45:27 UTC+01:00	UPDATE_COMPLETE, CANCELLED	AWS::EC2::Instance	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:45:27 UTC+01:00	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:45:27 UTC+01:00	P_IN_PROGRESS	AWS::EC2::Instance	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:45:23 UTC+01:00	UPDATE_COMPLETE	AWS::EC2::Instance	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:44:56 UTC+01:00	UPDATE_IN_PROGRESS	AWS::EC2::Instance	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:44:34 UTC+01:00	UPDATE_IN_PROGRESS	AWS::EC2::Instance	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:44:27 UTC+01:00	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2demostack	EC2 instance	ec2demostack
2014-06-23 13:39:56 UTC+01:00	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2demostack	User Initiated	ec2demostack
2014-06-23 13:37:54 UTC+01:00	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	ec2demostack	User Initiated	ec2demostack
2014-06-23 13:37:54 UTC+01:00	ROLLBACK_COMPLETE	AWS::EC2::Instance	ec2demostack	User Initiated	ec2demostack

At the bottom of the page, there are links for Feedback, Help, and Logout.

Using a template to create and manage a stack

The screenshot shows the CloudFormation Management Console interface. At the top, there are navigation links for CloudFormation Management, Services, VPC, EC2, RDS, CloudFront, and Help. Below the navigation bar, there are two sections: 'Events' and 'Stacks'.

Events: This section lists events for a specific stack named 'ec2InstanceDemoStack'. The events are ordered by time, from newest to oldest. The first event is highlighted with an orange border.

Time	Status	Type	Logical ID	Reason
2014-06-23 13:53:48 UTC+01:00	DELETE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceDemoStack	Ec2Instance
2014-06-23 13:53:45 UTC+01:00	DELETE_IN_PROGRESS	AWS::EC2::Instance		ec2InstanceDemoStack
2014-06-23 13:53:17 UTC+01:00	DELETE_IN_PROGRESS	AWS::EC2::Instance		User Initiated
2014-06-23 13:52:56 UTC+01:00	DELETE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	Ec2Instance
2014-06-23 13:45:53 UTC+01:00	UPDATE_COMPLETE	AWS::CloudFormation::Stack	ec2InstanceDemoStack	Ec2Instance
2014-06-23 13:45:53 UTC+01:00	DELETE_COMPLETE	AWS::EC2::Instance		Ec2Instance
2014-06-23 13:45:27 UTC+01:00	DELETE_IN_PROGRESS	AWS::EC2::Instance		User Initiated
2014-06-23 13:45:25 UTC+01:00	UPDATE_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	Ec2Instance
2014-06-23 13:45:23 UTC+01:00	UPDATE_COMPLETE	AWS::EC2::Instance		Ec2Instance
2014-06-23 13:44:35 UTC+01:00	UPDATE_IN_PROGRESS	AWS::EC2::Instance		Ec2Instance
2014-06-23 13:44:34 UTC+01:00	UPDATE_IN_PROGRESS	AWS::EC2::Instance		Ec2Instance
2014-06-23 13:44:27 UTC+01:00	UPDATE_IN_PROGRESS	AWS::CloudFormation::Stack	ec2InstanceDemoStack	Ec2Instance

Stacks: This section shows a single stack named 'ec2InstanceDemoStack'.

Stack Name	Created Time	Status	Description
ec2InstanceDemoStack	2014-06-23 13:44:27 UTC+01:00	CREATE_IN_PROGRESS	Creating stack ec2InstanceDemoStack

Using a template to create and manage a stack

The screenshot shows the AWS CloudFormation console with a stack named "Cleans Up Resources". The main interface displays a list of resources, including EC2 instances, VPCs, and Auto Scaling groups. A blue diagonal banner across the top left reads "Cleans Up Resources".

EC2 Instances:

Name	Instance ID	Type	Status
l-06072ec5	i-06072ec5	m3 medium	terminated
l-02e03792	i-02e03792	m3 medium	terminated

VPC:

VPC ID	AMIs	Public IP	Private IP
amzn-vpc-1	ami-049749c1	54.230.64.94 (ip2)	172.31.10.104 (ip1)

Auto Scaling Groups:

Launch Configuration	Instances	Health	Last Update
Launch Configuration 1	0	OK	2016-03-24T14:45:00Z

Logs:

CloudWatch Metrics Log Stream: /aws/lambda/CloudWatchLogsGetLogs

Feedback:

Leave Feedback

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon.com, the Amazon logo, and the Amazon.com logo are trademarks of Amazon.com, Inc. or its affiliates.

All other trademarks and copyrights are the property of their respective owners.

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-1fheliryvrdrbr-eu-west-1/2014174d0r-ec2Instance.template
  --parameters ParameterKey=KeyValuePair,ParameterValue=ManagementKeyValuePair
```

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 navigation links for Home, Services, CloudFront, WAF, VPC, EC2, Lambda, and Help. Below this is a search bar with filters for Active, By Name, Stack Name, Created Time, Status, and Description. A table lists stacks, with one entry for 'ec2Demo' highlighted by an orange border. The table columns include Overview, Outputs, Resources, Events, Template, Parameters, Type, Stack Policy, Logical ID, and Status Reason. The 'ec2Demo' entry has the following details:

Logical ID	Status Reason
ec2DemoCloudFrontLogDelivery	EC2InMaintenance
ec2DemoLambdaFunction	EC2InMaintenance
ec2DemoLambdaFunction	EC2InMaintenance
ec2DemoCloudFrontLogDelivery	User Initiated

At the bottom of the page, there is a note about copyright and terms of use, followed by Feedback and Report bugs buttons.

© 2006 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)

Feedback Report bugs

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-arms <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-arms <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

Template reference: docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/template-reference.html

```
"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

This screenshot shows the 'Specify Parameters' step of an AWS CloudFormation stack creation process. The top navigation bar includes 'Create A New Stack', 'Services' (selected), 'VPC', 'SS', 'EC2', 'S3', 'RDS', 'CloudFront', 'Edit', 'Ireland', and 'Help'. On the left, a sidebar has 'Select Template' and 'Specify Parameters' (selected). The main area has tabs for 'Options' and 'Review'. Below these are sections for 'Parameters' and 'ManagementKeyPair'. The 'ManagementKeyPair' section is highlighted with an orange border and contains the text: 'The EC2 Key Pair to allow SSH access to the instance'. At the bottom right are buttons for 'Cancel', 'Previous', 'Next', and 'Feedback'. The footer contains copyright information: '© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy' and 'Terms of Use'. The bottom right corner also features the text 'lesspack'.

Input Parameters

```
"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

```
{ "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 Pseudo parameters

[Fn::Base64](#)

[AWS::NotificationARNs](#)

[Fn::FindInMap](#)

[AWS::Region](#)

[Fn::GetAtt](#)

[AWS::StackId](#)

[Fn::GetAZs](#)

[AWS::StackName](#)

[Fn::Join](#)

[AWS::Region](#)

[Fn::Select](#)

[Ref](#)

Working with non-AWS Resources

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

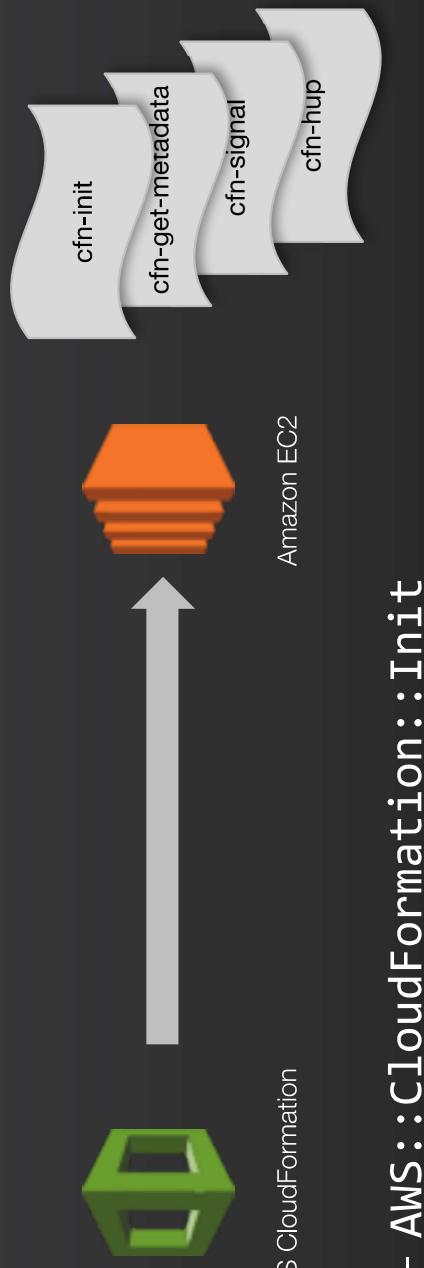
More on Custom Resources in 'AWS CloudFormation under the Hood' from re:Invent 2013: <http://youtu.be/ZhgMaw67yu0>
AWS CloudFormation Custom Resource Walkthrough documentation:
<docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/crpg-walkthrough.html>

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",
                "\n",
                "yum -y install gcc-c++ make",
                "\n",
                "yum -y install mysql-devel sqlite-devel",
                "\n",
                "yum -y install ruby-rdoc rubygems ruby-mysql ruby-devel",
                "\n",
                "gem install --no-ri --no-rdoc rails",
                "\n",
                "gem install --no-ri --no-rdoc mysql",
                "\n",
                "gem install --no-ri --no-rdoc sqlite3",
                "\n",
                "rails new myapp",
                "\n",
                "cd myapp",
                "\n",
                "rails server -d",
                "\n",
                "curl -X PUT -H 'Content-Type: application-binary' --data-binary '{\"Status\": \"SUCCESS\", \"Reason\": \"The application myapp is ready\", \"UniqueId\": \"myapp\", \"Data\": \"Done\"}'",
                "\n",
                {"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",
            "Comment3" : "The CloudFormationBootstrap sample script is run at the end of the bootstrap process"
        }
    },
    "AWS::CloudFormation::Interface" : {
        "Config" : {
            "Packages" : {
                "yum" : {
                    "mysql" : [],
                    "mysql-server" : [],
                    "mysql-libs" : [],
                    "httpd" : [],
                    "php" : [],
                    "php-mysql" : []
                }
            }
        }
    }
},
"sources" : {
    "/var/www/html" : "https://s3.amazonaws.com/cloudformation-examples/CloudFormationPHPSSample.zip"
}
```

Installing & executing CloudFormation helper

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" : "Region" }, " || error_exit 'Failed to run cfn-init'\n"
    ] ] } }
},
```

Creating files on Instance Filesystems

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  
        ],  
        "mode" : "000644",  
        "owner" : "root",  
        "group" : "root"  
    }  
}
```

Controlling Services

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"  
        }  
    }  
}
```

More on Deploying Applications with AWS CloudFormation:
docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/deploying.applications.html

Yes!

All this functionality is available for
Windows instances too!

Bootstrapping AWS CloudFormation Windows Stacks:
docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/cfn-windows-stacks-bootstrapping.html

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.

What about Chef? and/or

What about Puppet?

Find out more here: aws.amazon.com/cloudformation/aws-cloudformation-articles-and-tutorials/

Bootstrapping Applications via AWS CloudFormation

AWS CloudFormation gives you an easy way to create the set of resources such as Amazon EC2 instances, Amazon RDS database instances and Elastic Load Balancers needed to run your application. This 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 leaves the obvious question of how your application software is deployed, configured and monitored 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... \[?\]](#)

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... \[?\]](#)

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 managing, configuring and patching application 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... \[?\]](#)

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 fncluster HPC deployment framework:

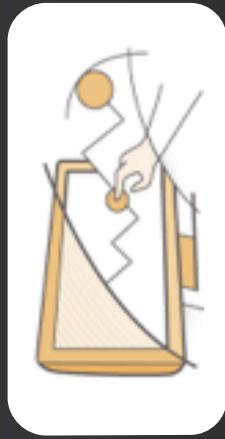
github.com/aws-labs/cfncluster/

CloudFormation Template to Deploy Wordpress

<https://s3-us-west-1.amazonaws.com/cloudformation-templates-us-west-1/WordPress/Multi-AZ-template.zip>

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)

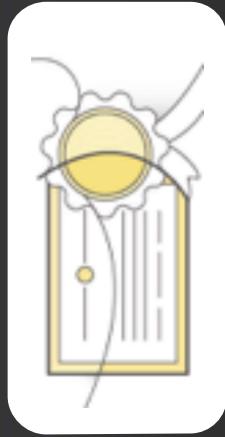
Training



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



AWS Summit

LONDON

15 APRIL 2015

aws.amazon.com/summits/london/

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





amazon web services

Ian Massingham – Technical Evangelist

 @IanMmmmm

 @AWS_UK1 for local AWS events & news

 @AWScloud for Global AWS News & Announcements

FOLLOW US FOR MORE
Events & Webinars