



**-By Sriganesh Pera**

### **AWS Cloudformation:**

AWS Cloudformation lets you provision all AWS services using a piece of code, which is why it is called as infrastructure as a code. That code can be written in JSON/YAML.

CloudFormation has two parts: **templates and stacks**. A template is a JavaScript Object Notation (JSON) text file. This file defines what AWS resources or non-AWS resources are required to run the application.

For example, the template may declare that the application requires an Amazon Elastic Compute Cloud (EC2) instance and an Identity and Access Management (IAM) policy.

When the template is submitted to the service, CloudFormation creates the necessary resources in the customer's account and builds a running instance of the template, putting dependencies and data flows in the right order automatically. The running instance is called a stack.

### **Cloudformation Concepts:**

#### **1. Stack:**

A stack is Cloud Formation unit for grouping infrastructure. Infrastructure refers to group of AWS resources: EC2 instances, ELB, S3 buckets and RDS database instance etc., Stack controls the life cycle of the infrastructure -creating, updating, deleting infrastructure resources.

## **2. Change sets:**

A change set consists of summary of AWS resources that are grouped under for reference before updating the stack. Change sets allow you to see how your changes might impact your running resources, especially for critical resources, before implementing them.

For example, if you change the name of an Amazon RDS database instance, AWS CloudFormation will create a new database and delete the old one. You will lose the data in the old database unless you've already backed it up. If you generate a change set, you will see that your change will cause your database to be replaced, and you will be able to plan accordingly before you update your stack.

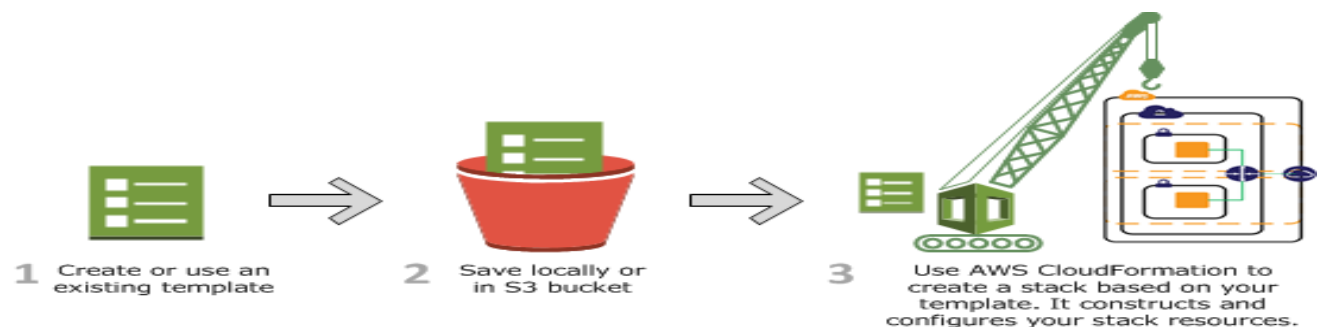
## **3. Template:**

A JSON document which defines the creation of AWS resources. A template is the body of a stack. When a stack gets created all the resources defined in a template gets executed.

## **4. Stack policy:**

An IAM policy statement that tells what can be changed in relation to stack and by who. A policy can be added during creation time, or it can be added to existing stack afterwards. But once a stack policy is added it cannot be deleted.

### **CloudFormation workflow:**



**Step 1:** You create a template in JSON, which defines the required AWS resources to launch

**Step2:** Save that JSON document either in local drive or in S3

**Step3:** Upload the JSON document to create stack, which takes care of provisioning AWS resources.

What does this template consists of?

As said earlier a template is written in JSON document. Let's see the anatomy of the template.

1. **Parameters:** allows a way to inject info. into the template  
that info could be: EC2 instance types, AMI ID, IP address range and SSH keys

**syntax:**

JSON

```
"Parameters" : {  
  "InstanceTypeParameter" : {  
    "Type" : "String",  
    "Default" : "t2.micro",  
    "AllowedValues" : ["t2.micro", "m1.small", "m1.large"],  
    "Description" : "Enter t2.micro, m1.small, or m1.large. Default is t2.micro."  
  }  
}
```

2. **Mappings:** It allows hashing's or dictionaries to be defined in a template. The Mappings section matches a key to a corresponding set of named values just like a key value pair.

**Syntax:**

```
"Mappings" : {  
  "RegionMap" : {  
    "us-east-1" : { "32" : "ami-6411e20d"},  
    "us-west-1" : { "32" : "ami-c9c7978c"},  
    "eu-west-1" : { "32" : "ami-37c2f643"},  
    "ap-southeast-1" : { "32" : "ami-66f28c34"},  
    "ap-northeast-1" : { "32" : "ami-9c03a89d"}  
  }  
}
```

3. **Resources:** This is where actual AWS resources are declared. This is a mandatory section as rest are all optional.

**Syntax:**

```
"Resources" : {  
  "MyEC2Instance" : {  
    "Type" : "AWS::EC2::Instance",  
    "Properties" : {  
      "ImageId" : "ami-2f726546"  
    }  
  }  
}
```

#### **4. Outputs:**

The Outputs section declares output values. It tells where the results from the templates can be stored or it tells how the information can be presented based on conditional logic.

**Syntax:**

```
"Outputs" : {  
  "BackupLoadBalancerDNSName" : {  
    "Description": "The DNSName of the backup load balancer",  
    "Value" : { "Fn::GetAtt" : [ "BackupLoadBalancer", "DNSName" ] },  
    "Condition" : "CreateProdResources"  
  },  
  "InstanceID" : {  
    "Description": "The Instance ID",  
    "Value" : { "Ref" : "EC2Instance" }  
  }  
}
```

In the above example, the output named BackupLoadBalancerDNSName returns the DNS name for the resource with the logical ID BackupLoadBalancer only when the CreateProdResources condition is true.

## **Lab 1 : Intro to Cloud Formation**

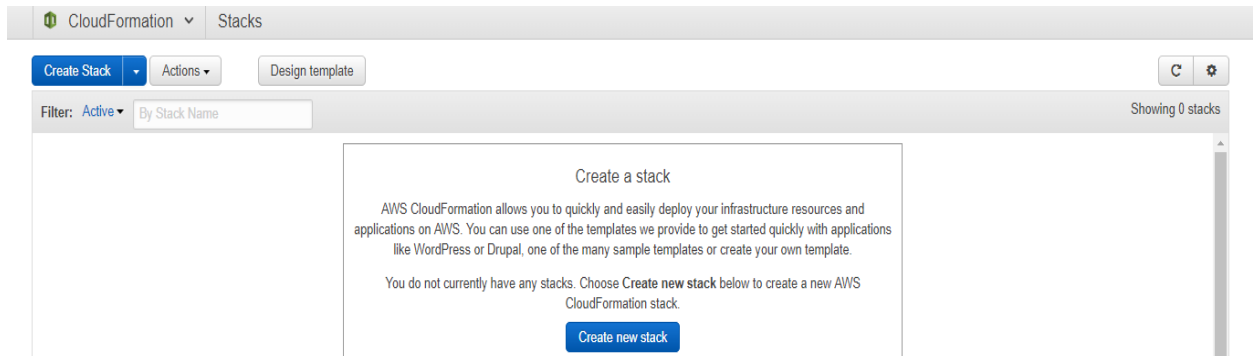
### **Step1: Download the cloudformation template from my github**

<https://github.com/perasr/cloudformation.git>

use the wordpress template

### **Step 2: Create a cloudformation stack**

Go to Cloudformation in AWS Dashboard and click on it. Then click on create new stack.



Now select 'upload a template using Amazon S3' and then locate the file that you have just downloaded from github account, and then click next.

The screenshot shows the 'Create Stack' wizard in the AWS CloudFormation console. The breadcrumb navigation at the top reads 'CloudFormation > Stacks > Create Stack'. The main heading is 'Create stack'. On the left, there is a sidebar with links: 'Select Template' (active), 'Specify Details', 'Options', and 'Review'. The main content area is titled 'Select Template' and includes a sub-header 'Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.' Below this, there are two sections: 'Design a template' with a 'Design template' button, and 'Choose a template' with three radio button options: 'Select a sample template' (with a dropdown menu), 'Upload a template to Amazon S3' (selected, with a 'Choose File' button and the filename 'wordpress.template'), and 'Specify an Amazon S3 template URL' (with an empty text input field). At the bottom right, there are 'Cancel' and 'Next' buttons.

### Step 3: Define the parameters for the Template

Now fill the stack name and parameters as shown:

Constraints for DBPassword and DBRootPassword

**MinLength: "8", "MaxLength": "41" "AllowedPattern" : "[a-zA-Z0-9]\*"**

**And the Password should start with a letter. Also, ensure that passwords for DBPassword and DBRootPassword are not the same i.e. different.**

The screenshot shows the 'Define parameters' step of the 'Create Stack' wizard. The 'Stack name' field is filled with 'WordPress'. Below it, the 'Parameters' section contains several input fields with their respective descriptions: 'DBName' (wordpressdb, 'The WordPress database name'), 'DBPassword' (masked with asterisks, 'The WordPress database admin account password'), 'DBRootPassword' (masked with asterisks, 'MySQL root password'), 'DBUser' (masked with asterisks, 'The WordPress database admin account username'), 'InstanceType' (t2.small, 'WebServer EC2 instance type'), 'KeyName' (qwikLABS-L198-1972039, 'Name of an existing EC2 KeyPair to enable SSH access to the instances'), and 'SSHLocation' (0.0.0.0/0, 'The IP address range that can be used to SSH to the EC2 instances'). At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

## So what have you done so far?

You've uploaded a template written in .json format. That template consists of info. about AWS resources needed to launch a WordPress web site on a single instance.

AWS uses the info in the template to launch the resources needed to launch Wordpress website.

What are all the AWS resources that this template covers?

1. Ec2 instances and 3 security groups.
2. SSH keys
3. Apache Web server installation
4. Wordpress installation
5. Creates user data: Username,Password,and Database name for admin purposes.

## Step 4: Review and launch

[Select Template](#)  
[Specify Details](#)  
[Options](#)  
[Review](#)

Review

Template

Template URL

<https://s3-us-west-2.amazonaws.com/cf-templates-1ogg67smfylx5-us-west-2/20173380a6-wordpress.template>

Description

AWS CloudFormation Sample Template WordPress\_Single\_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. **\*\*WARNING\*\*** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Estimate cost

[Cost](#)

Details

Stack name:

WordPress

DBName

wordpressdb

DBPassword

.....

DBRootPassword

.....

DBUser

.....

InstanceType

t2.small

KeyName

qwikLABS-L198-1972039

SSHLocation

0.0.0.0/0

Options

Tags

No tags provided

Advanced

Notification

Termination Protection

Disabled

As you can see the AWS cloudformation stack gets created and you can find in the description about the Stack that is created.

CloudFormation > Stacks > Stack Detail

### WordPress

Other Actions Update Stack

Stack name: WordPress

Stack ID: arn:aws:cloudformation:us-west-2:442928696750:stack/WordPress/391e0850-d8b1-11e7-95fe-503ac9316835

Status: CREATE\_COMPLETE

Status reason:

Termination protection: Disabled

IAM role:

Description: AWS CloudFormation Sample Template WordPress\_Single\_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. **\*\*WARNING\*\*** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

## Step 5: Monitoring the template creation and understanding the process

### (a) Templates:

Now click on the template, you notice that the info about the AWS resources (Ec2 instances, instance types, installation of Apache webserver etc.,) that are provisioned for the launch of WordPress website.

Template

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. **WARNING** This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {

    "KeyName": {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
      "Type": "AWS::EC2::KeyPair::KeyName",
      "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
    },
  },

  "InstanceType" : {
```

### (b) Resources:

You can see what are all the resources deployed for the Wordpress website as specified in the template.

Resources

Logical ID	Physical ID	Type	Status	Status Reason
WebServer	i-0561563bb5cd20fd1	AWS::EC2::Instance	CREATE_COMPLETE	
WebServerSecurityGroup	WordPress-WebServerSecurityGroup-18941NPTYM2JZ	AWS::EC2::SecurityGroup	CREATE_COMPLETE	

### (c) Events:

Click on the events tab to monitor the progress of stack creation. As you notice all the AWS resources needed for the stack to keep it up and running and being created.



▼ Events

2017-12-04	Status	Type	Logical ID	Status Reason
▶ 10:40:21 UTC+0550	CREATE_COMPLETE	AWS::CloudFormation::Stack	WordPress	
▶ 10:40:17 UTC+0550	CREATE_COMPLETE	AWS::EC2::Instance	WebServer	
▶ 10:40:17 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	WebServer	Received SUCCESS signal with UniquelId I-0561563bb5cd20fd1
▶ 10:39:03 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	WebServer	Resource creation Initiated
▶ 10:39:02 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	WebServer	
▶ 10:38:59 UTC+0550	CREATE_COMPLETE	AWS::EC2::SecurityGroup	WebServerSecurityGroup	
▶ 10:38:58 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::SecurityGroup	WebServerSecurityGroup	Resource creation Initiated
▶ 10:38:57 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::SecurityGroup	WebServerSecurityGroup	
▶ 10:38:53 UTC+0550	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	WordPress	User Initiated

### (d) Outputs:

You will find the value of the web URL for the stack that is just created. Click on it and You will see the Wordpress website populated:

▼ Outputs

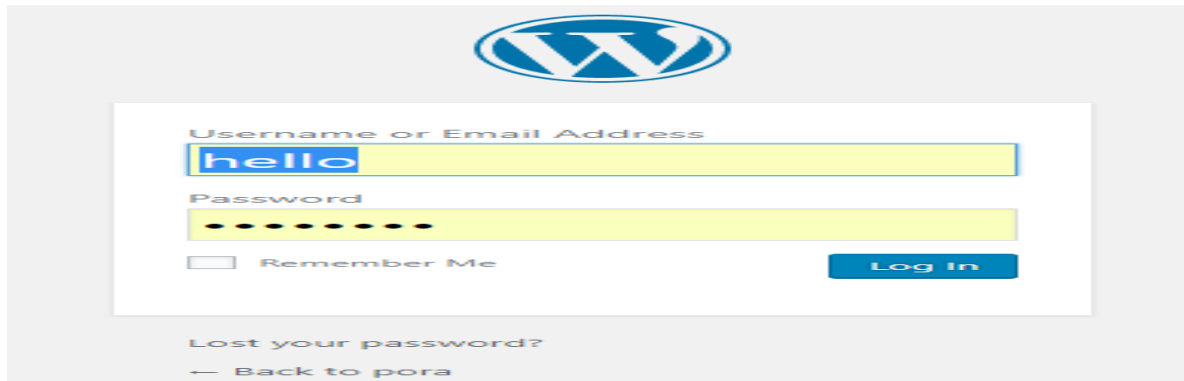
Key	Value	Description	Export Name
WebsiteURL	<a href="http://ec2-35-164-214-173.us-west-2.compute.amazonaws.com/wordpress">http://ec2-35-164-214-173.us-west-2.compute.amazonaws.com/wordpress</a>	WordPress Website	

You will see the welcome page for the Wordpress website. Enter the title name, Username and Password.

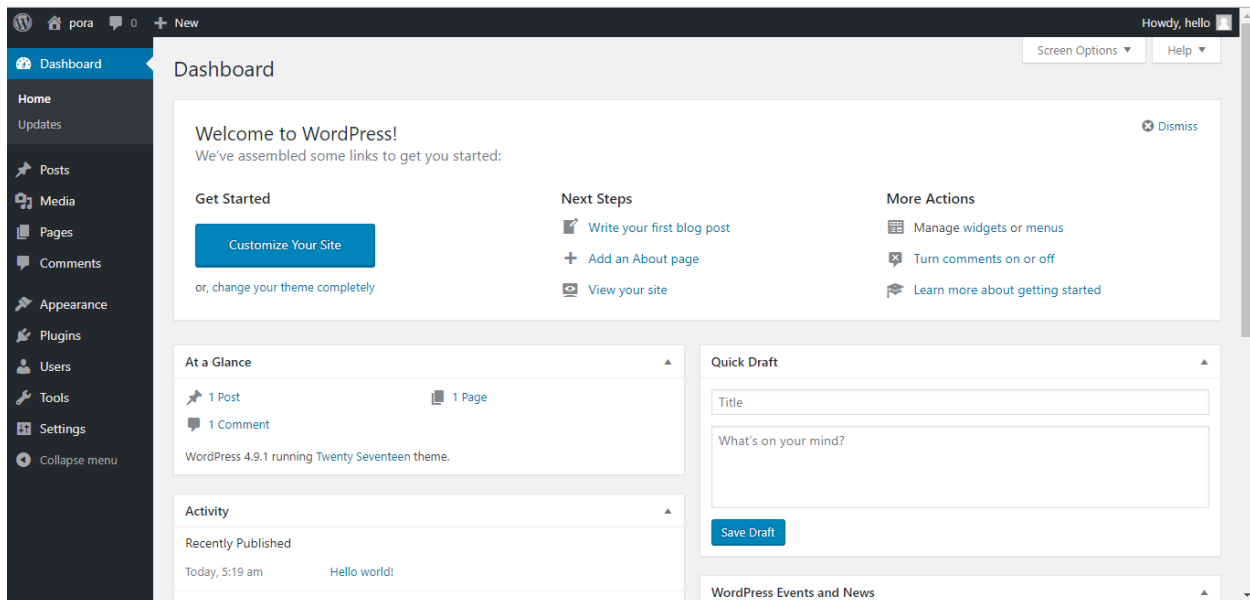
The Username and Password is of your choice.

It **need not** reflect the DBPassword or DBUsername that you entered as parameters in the STEP 3.

After you've entered the details then click install and after installation log in to you Wordpress using the username and password

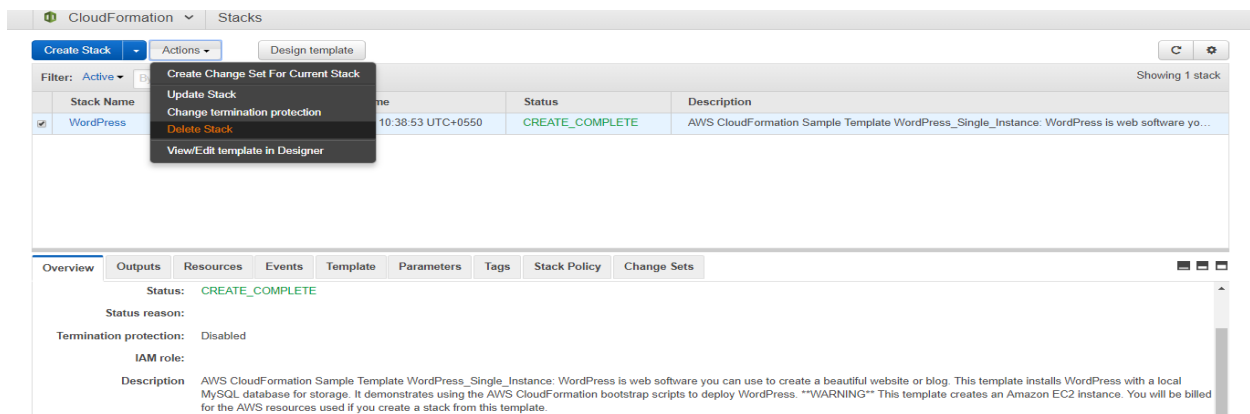


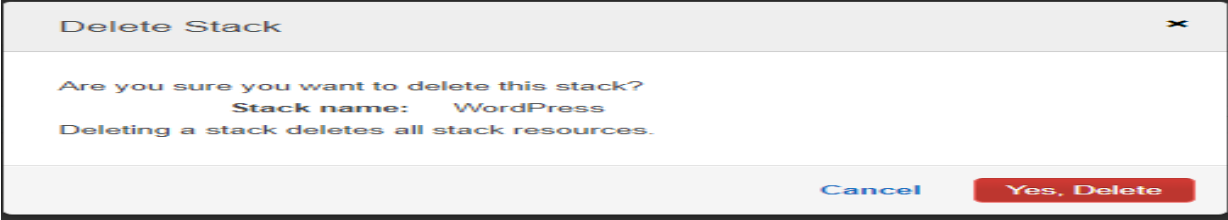
Now you're taken to your WordPress blog post. Feel free to play around.



## Step 6 :Clean up!

### Delete stack:





Monitor the progress of deletion:

Overview

Outputs

Resources

Events

Template

Parameters

Tags

Stack Policy

Change Sets

Status: DELETE\_IN\_PROGRESS

Status reason: User Initiated

Termination protection: Disabled

IAM role:

Description: AWS CloudFormation Sample Template WordPress\_Single\_Instance: WordPress is web software you can use to create a beautiful website or blog. This template installs WordPress with a local MySQL database for storage. It demonstrates using the AWS CloudFormation bootstrap scripts to deploy WordPress. \*\*WARNING\*\* This template creates an Amazon EC2 instance. You will be billed for the AWS resources used if you create a stack from this template.

Overview	Outputs	Resources	Events	Template	Parameters	Tags	Stack Policy	Change Sets
2017-12-04								
		Status	Type	Logical ID			Status Reason	
▶ 10:56:20 UTC+0550		DELETE_IN_PROGRESS	AWS::EC2::Instance	WebServer			User Initiated	
▶ 10:56:18 UTC+0550		DELETE_IN_PROGRESS	AWS::CloudFormation::Stack	WordPress				
▶ 10:40:21 UTC+0550		CREATE_COMPLETE	AWS::CloudFormation::Stack	WordPress			Received SUCCESS signal with UniqueId i-0561563bb5cd20fd1 Resource creation Initiated	
▶ 10:40:17 UTC+0550		CREATE_COMPLETE	AWS::EC2::Instance	WebServer				
▶ 10:40:17 UTC+0550		CREATE_IN_PROGRESS	AWS::EC2::Instance	WebServer				
▶ 10:39:03 UTC+0550		CREATE_IN_PROGRESS	AWS::EC2::Instance	WebServer				
▶ 10:39:02 UTC+0550		CREATE_IN_PROGRESS	AWS::EC2::Instance	WebServer			Resource creation Initiated	
▶ 10:38:59 UTC+0550		CREATE_COMPLETE	AWS::EC2::SecurityGroup	WebServerSecurityGroup				
▶ 10:38:58 UTC+0550		CREATE_IN_PROGRESS	AWS::EC2::SecurityGroup	WebServerSecurityGroup			Resource creation Initiated	
▶ 10:38:57 UTC+0550		CREATE_IN_PROGRESS	AWS::EC2::SecurityGroup	WebServerSecurityGroup				

CloudFormation

Stacks

Create Stack

Actions

Design template

Filter: Active

By Stack Name

Showing 1 stack

Stack Name	Created Time	Status	Description
<input checked="" type="checkbox"/> WordPress	2017-12-04 10:38:53 UTC+0550	CREATE_IN_PROGRE...	AWS CloudFormation Sample Template WordPress_Single_Instance: WordPress is web software yo...

Overview

Outputs

Resources

Events

Template

Parameters

Tags

Stack Policy

Change Sets

2017-12-04

▶ 10:38:53 UTC+0550

Status

CREATE\_IN\_PROGRESS

Type

AWS::CloudFormation::Stack

Logical ID

WordPress

Status Reason

User Initiated

~~~~~End of Lab 1~~~~~

## **Lab 2: AWS CloudFormation Deep Dive**

### **Step 1: Create a Stack**

Go to AWS dashboard and click on CloudFormation and click on create stack and click next.

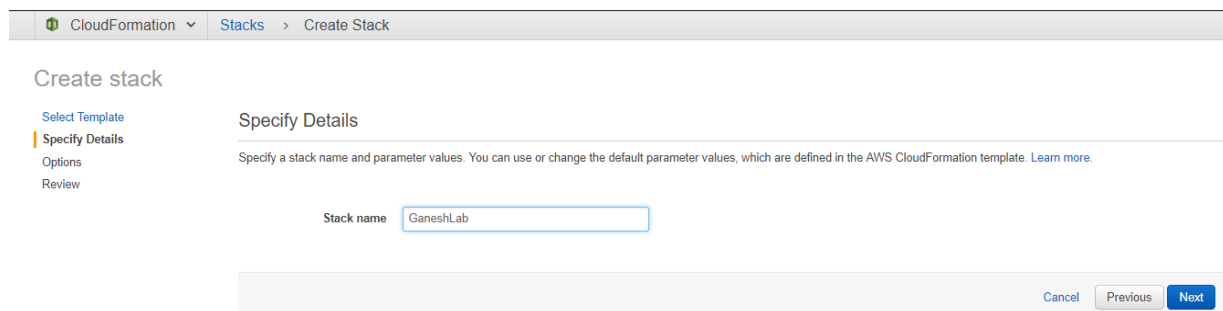
### **Step 2: Upload file to S3**

<https://github.com/perasr/cloudformation.git>

Use the templatefors3bucket template

In the create stack window, in the 'Select Template' page. Click on upload a template from Amazon S3. Chose the template that you've downloaded and click next.

In the specify details ,provide any name for your stack



The screenshot shows the AWS CloudFormation console's 'Create Stack' wizard. The breadcrumb navigation at the top reads 'CloudFormation > Stacks > Create Stack'. On the left, a sidebar lists the steps: 'Select Template' (with a blue link), 'Specify Details' (highlighted with an orange bar), 'Options', and 'Review'. The main content area is titled 'Specify Details' and includes a sub-header 'Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)'. Below this, there is a 'Stack name' label and a text input field containing 'GaneshLab'. At the bottom right, there are three buttons: 'Cancel', 'Previous' (disabled), and 'Next' (active).

Options page allow you to specify Tags, permissions and Advanced options. Use the default settings and click Next.

Now review the template and click Create.

Create stack

Select Template  
Specify Details  
Options  
**Review**

**Review**

**Template**

Template URL: <https://s3-us-west-2.amazonaws.com/cf-templates-xki48n87iyug-us-west-2/2017340TSG-lab1.template>  
Description: Basic S3 Bucket CloudFormation template  
Estimate cost: Cost

**Details**

Stack name: GaneshLab

**Options**

Tags  
No tags provided

Advanced

Notification: Disabled  
Termination Protection: none  
Timeout: none  
Rollback on failure: Yes

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

Cancel Previous **Create**

Now you can monitor the status of your stack creation by clicking on Events:

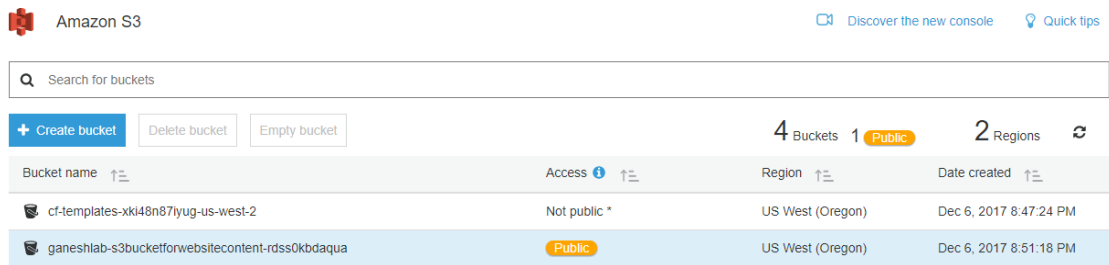
| CloudFormation ▾                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                              | Stacks              |                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------|-----------------------------------------|
| Create Stack ▾                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                              | Actions ▾           | Design template                         |
| Filter: Active ▾ By Stack Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                              | Showing 2 stacks    |                                         |
| Stack Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Created Time                 | Status              | Description                             |
| <input checked="" type="checkbox"/> GaneshLab                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 2017-12-06 20:51:12 UTC+0550 | CREATE_IN_PROGRE... | Basic S3 Bucket CloudFormation template |
| <input type="checkbox"/> qls-1978726-a73a1ae56850eb2f                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 2017-12-06 20:33:43 UTC+0550 | CREATE_COMPLETE     |                                         |
| <div> <div>Overview</div> <div>Outputs</div> <div>Resources</div> <div>Events</div> <div>Template</div> <div>Parameters</div> <div>Tags</div> <div>Stack Policy</div> <div>Change Sets</div> </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                              |                     |                                         |
| <div> <div>2017-12-06</div> <div> <div>20:51:40 UTC+0550</div> <div>20:51:38 UTC+0550</div> <div>20:51:17 UTC+0550</div> <div>20:51:16 UTC+0550</div> <div>20:51:12 UTC+0550</div> </div> <div> <div>CREATE_COMPLETE</div> <div>CREATE_COMPLETE</div> <div>CREATE_IN_PROGRESS</div> <div>CREATE_IN_PROGRESS</div> <div>CREATE_IN_PROGRESS</div> </div> <div> <div>Type</div> <div>AWS::CloudFormation::Stack</div> <div>AWS::S3::Bucket</div> <div>AWS::S3::Bucket</div> <div>AWS::S3::Bucket</div> <div>AWS::CloudFormation::Stack</div> </div> <div> <div>Logical ID</div> <div>GaneshLab</div> <div>S3BucketForWebsiteContent</div> <div>S3BucketForWebsiteContent</div> <div>S3BucketForWebsiteContent</div> <div>GaneshLab</div> </div> <div> <div>Status Reason</div> <div></div> <div>Resource creation Initiated</div> <div></div> <div>User Initiated</div> </div> </div> |                              |                     |                                         |

After a minute or so you will see the status of the stack:Create\_Complete

| CloudFormation ▾                              |                              | Stacks           |                                         |
|-----------------------------------------------|------------------------------|------------------|-----------------------------------------|
| Create Stack ▾                                |                              | Actions ▾        | Design template                         |
| Filter: Active ▾ By Stack Name                |                              | Showing 2 stacks |                                         |
| Stack Name                                    | Created Time                 | Status           | Description                             |
| <input checked="" type="checkbox"/> GaneshLab | 2017-12-06 20:51:12 UTC+0550 | CREATE_COMPLETE  | Basic S3 Bucket CloudFormation template |

### Step 3: View the bucket

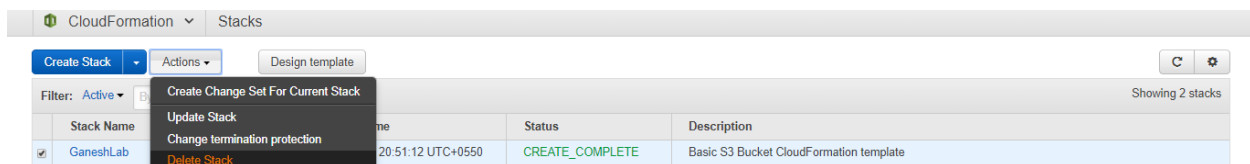
Now go to S3 and view the bucket that is created. It starts with the Stack name, in my case it's **Ganesh**



Above the bucket you will see another S3 Bucket created by CloudFormation. That bucket consists of the file you have uploaded.

#### **Step 4: Delete the stack**

Now go to AWS CloudFormation dashboard, then select the stack and then click actions and delete stack.



With the deletion of Stack, my bucket also gets deleted. You can monitor your stack deletion in the events tab as well.

#### **Step 5: Retaining the resources in S3 Bucket**

Basically, this template that you're about to download is the same as the previous one but with one change. There's an added property in that template "Deletion Policy": "Retain", which will retain the AWS resources specified in the template, in this case it retains S3 bucket even after deletion of the stack.

Now download this template and create a stack with any name and upload this file as a template and after the stack gets created, delete the stack.

<https://github.com/perasr/cloudformation.git>

use the retains3.template

After it gets deleted, go back to S3 and you will find that the bucket that is created due to the AWS CloudFormation stack still remains.

| + Create bucket                                   |              | Delete bucket    | Empty bucket           | 4 Buckets 1 Public |  | 2 Regions | ↻ |
|---------------------------------------------------|--------------|------------------|------------------------|--------------------|--|-----------|---|
| Bucket name                                       | Access       | Region           | Date created           |                    |  |           |   |
| cf-templates-xki48n87iyug-us-west-2               | Not public * | US West (Oregon) | Dec 6, 2017 8:47:24 PM |                    |  |           |   |
| ganeshlab2-s3bucketforwebsitecontent-lzxc6688dcda | Public       | US West (Oregon) | Dec 6, 2017 9:24:15 PM |                    |  |           |   |

Ganesh Lab 2 is the bucket that is created when CloudFormation stack is created. Even after deletion of stack that bucket still remains there because in the template 'Delete Policy: Retain' is specified. So, in this way you keep your resources even after deletion of stack.

### **Step 6: Deploy a PHP web app using Aws CloudFormation**

Now download the template. Examine the template and you will notice that this template provisions EC2 instances and installs Apache Webserver and PHP and hosts a single instance PHP application. Installation is done through AWS helper scripts.

<https://github.com/perasr/cloudformation.git>

use the template phpstack.template

After stack creation we will be manipulating(or uploading) many resources in the stack. There is a daemon called cfn-hup, which listens to the changes made to the configuration of template defined in the metadata of EC2 instance. By using cfn-hup daemon, you can update application software, such as version of PHP or Apache or you can update the application file itself from AWS Cloudformation.

If you notice the AWS cloudformation template there is section called `etc/cfn/hooks.d` in there AWS CloudFormation triggers the `cfn-auto-reloader.conf` hooks file when you change the `AWS::CloudFormation::Init` resource that is associated with the `Webserver` resource.

### **Step 7: Create the stack using the template**

Now click on launch stack and upload the file from your local drive, then it will take you to specify parameters page, where you're asked to give the parameters like VPC ID and subnet ID.

You either create a new VPC and new subnet and provide the ID or you can use the default one, and select instance t2.micro.

## Create stack

Select Template  
Specify Details  
Options  
Review

### Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

### Parameters

VPCId  VPC ID of the Lab VPC

VPCPublicSubnetId  Subnet ID of the public subnet within the Lab VPC

WebServerInstanceType  WebServer EC2 instance type

In my case, I actually created a new VPC and subnet in US-West2a region, and supplied those details as shown in the above picture.

Now click next and then comes the options page, then click next. On the review page, review your configuration and click create.

| CloudFormation                                 |                              | Stacks              |                                                                                                    |
|------------------------------------------------|------------------------------|---------------------|----------------------------------------------------------------------------------------------------|
| Create Stack                                   | Actions                      | Design template     |                                                                                                    |
| Filter: Active                                 | By Stack Name                |                     | Showing 2 stacks                                                                                   |
| Stack Name                                     | Created Time                 | Status              | Description                                                                                        |
| <input checked="" type="checkbox"/> ganeshlab3 | 2017-12-09 05:46:53 UTC+0550 | CREATE_IN_PROGRE... | AWS CloudFormation Sample Template Update Tutorial Part 1: Sample template that can be used to ... |

As you can see, the ganeshlab3 is getting created. It will take at least 5 minutes for stack creation. After the stack creation is completed, check the outputs tab you will find the WebsiteURL. Copy and paste it in a new browser and hit enter and you will see this:

### AWS CloudFormation sample PHP application

An elastic IP is associated for this URL as defined in the template.

### Step 8: Updating the stack.

If you notice the Mappings section of the template you will see that there are 2 instance types listed t2.micro and m4.large. Now, I can easily switch the m4.large by simply updating the stack.



```

24     "Mappings": {
25         "AWSInstanceType2Arch": {
26             "t2.micro": {
27                 "Arch": "64"
28             },
29             "m4.large": {
30                 "Arch": "64"
31             }

```

Click on the actions button of your stack and click update stack. Now this will take you to Templates page, click next and you will see specify details page, and under parameters change the instance type to m4.large.

Parameters

VPCId:  VPC ID of the Lab VPC

VPCPublicSubnetId:  Subnet ID of the public subnet within the Lab VPC

WebServerInstanceType:  WebServer EC2 instance type

Now click next and then next again, this will take you to review page where you can preview the change

Preview your changes

Based on your input, CloudFormation will change the following resources. For more information, choose [View change set details](#).

| Action                 | Logical ID    | Physical ID         | Resource type      | Replacement |
|------------------------|---------------|---------------------|--------------------|-------------|
| <a href="#">Modify</a> | Endpoint      | 54.148.197.40       | AWS::EC2::EIP      | False       |
| <a href="#">Modify</a> | WebServerHost | i-0c4a725c95d263d18 | AWS::EC2::Instance | Conditional |

Now click update. Now you can see the progress of your updates.

CloudFormation ▾ Stacks

[Create Stack](#) ▾ [Actions](#) ▾ [Design template](#)

Filter: [Active](#) ▾ | By Stack Name

Showing 2 stacks

| Stack Name                                            | Created Time                 | Status               | Description                                                                                        |
|-------------------------------------------------------|------------------------------|----------------------|----------------------------------------------------------------------------------------------------|
| <input checked="" type="checkbox"/> ganeshlab3        | 2017-12-09 05:46:53 UTC+0550 | UPDATE_IN_PROGRES... | AWS CloudFormation Sample Template Update Tutorial Part 1: Sample template that can be used to ... |
| <input type="checkbox"/> qls-1984781-a36ec054b922a9f6 | 2017-12-09 04:48:57 UTC+0550 | CREATE_COMPLETE      |                                                                                                    |

Overview Outputs Resources **Events** Template Parameters Tags Stack Policy Change Sets

| 2017-12-09        | Status             | Type                               | Logical ID             | Status Reason               |
|-------------------|--------------------|------------------------------------|------------------------|-----------------------------|
| 06:04:25 UTC+0550 | UPDATE_IN_PROGRESS | AWS::CloudFormation::Stack         | ganeshlab3             | User Initiated              |
| 05:49:57 UTC+0550 | CREATE_COMPLETE    | AWS::EC2::EIP                      | Endpoint               |                             |
| 05:49:54 UTC+0550 | CREATE_COMPLETE    | AWS::CloudFormation::WaitCondition | WebServerWaitCondition | Resource creation Initiated |
| 05:49:52 UTC+0550 | CREATE_IN_PROGRESS | AWS::EC2::EIP                      | Endpoint               |                             |
| 05:49:22 UTC+0550 | CREATE_IN_PROGRESS | AWS::CloudFormation::WaitCondition | WebServerWaitCondition | Resource creation Initiated |
| 05:49:22 UTC+0550 | CREATE_IN_PROGRESS | AWS::CloudFormation::WaitCondition | WebServerWaitCondition |                             |
| 05:49:21 UTC+0550 | CREATE_IN_PROGRESS | AWS::EC2::EIP                      | Endpoint               |                             |
| 05:49:19 UTC+0550 | CREATE_COMPLETE    | AWS::EC2::Instance                 | WebServerHost          |                             |
| 05:49:02 UTC+0550 | CREATE_IN_PROGRESS | AWS::EC2::Instance                 | WebServerHost          | Resource creation Initiated |

## Step 9: Making your application highly available.

Create an IAM role named 'ec2-role-2' (make sure you give the same name as specified here). Attach following policies as shown:

Attach policy

Attached policies: 2

| Policy name                                                                                                       | Policy type        |   |
|-------------------------------------------------------------------------------------------------------------------|--------------------|---|
|  AmazonEC2ReadOnlyAccess         | AWS managed policy | ✕ |
|  AWSCloudFormationReadOnlyAccess | AWS managed policy | ✕ |

Add inline policy

Now use the template autoscaling.template that you’ve downloaded from my github.Then click next, You will see the specify details page. In that for the PublicSubnetAZ give the value as us-west-2a.

Stack name ganeshlab3

Parameters

VPCId vpc-f8b15181 VPC ID of the Lab VPC

VPCPublicSubnetAZ us-west-2a AZ of the public subnet within the Lab VPC

VPCPublicSubnetId subnet-9c5f73fa Subnet ID of the public subnet within the Lab VPC

WebServerInstanceType m4.large WebServer EC2 instance type

In the preview page you will see this:

Preview your changes

Based on your input, CloudFormation will change the following resources. For more information, choose [View change set details](#).

| Action | Logical ID               | Physical ID                                       | Resource type                           | Replacement |
|--------|--------------------------|---------------------------------------------------|-----------------------------------------|-------------|
| Add    | ElasticLoadBalancer      |                                                   | AWS::ElasticLoadBalancing::LoadBalancer |             |
| Remove | Endpoint                 | 54.148.197.40                                     | AWS::EC2::EIP                           |             |
| Add    | WebServerConfig          |                                                   | AWS::AutoScaling::LaunchConfiguration   |             |
| Add    | WebServerGroup           |                                                   | AWS::AutoScaling::AutoScalingGroup      |             |
| Remove | WebServerHost            | i-0c4a725c95d263d18                               | AWS::EC2::Instance                      |             |
| Modify | WebServerInstanceProfile | ganeshlab3-WebServerInstanceProfile-1UET9EQB6U0OL | AWS::IAM::InstanceProfile               | False       |

Cancel

Previous

Update

## AWS CloudFormation sample PHP application

Highly available, load balanced stack updated version via UpdateStack