

## Module 5: Configuration Management Using OpsWorks

---

### Demo Document 2

edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Automating Deployment In Opsworks Stack Using CodePipeline

### Demo steps:

#### Step 1: Create a AWS Opsworks Environment and CodePipeline through AWS CloudFormation

- Type the below code in your local system and save it in the name of Codepipeline-opswork.json

```
{
  "AWSTemplateFormatVersion":"2010-09-09",
  "Description":"OpsWorks stack and Deployment Pipeline in CodePipeline",
  "Parameters":{
    "KeyName":{
      "Description":"Name of an existing EC2 KeyPair for SSH access to the instances.",
      "Type":"AWS::EC2::KeyPair::KeyName"
    },
    "RepositoryName":{
      "Description":"The name of the CodeCommit repo",
      "Default":"opsworks-php-demo",
      "Type":"String",
      "AllowedPattern":"[\\x20-\\x7E]*",
      "ConstraintDescription":"Can contain only ASCII characters."
    },
    "RepositoryBranch":{
      "Description":"The name of the branch for the CodeCommit repo",
      "Type":"String",
      "Default":"master",
      "AllowedPattern":"[\\x20-\\x7E]*",
      "ConstraintDescription":"Can contain only ASCII characters."
    }
  },
  }
```

```
"Metadata":{"AWS::CloudFormation::Interface":{
  "ParameterGroups":[
    { "Label":{
      "default":"Dynamic Configuration"
    },
    "Parameters":["KeyName"
    ] },
    { "Label":{"default":"CodeCommit Configuration"
    },
    "Parameters":["RepositoryName",
      "RepositoryBranch"
    ] }}
  },
  "Mappings":{
    "Region2Principal":{
      "us-east-1":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com" },
      "us-west-2":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com"},
      "us-west-1":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com"},
      "eu-west-1":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com" },
      "ap-southeast-1":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com" },
      "ap-northeast-1":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com" },
      "ap-northeast-2":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com" },
      "ap-southeast-2":{"EC2Principal":"ec2.amazonaws.com",
        "OpsWorksPrincipal":"opsworks.amazonaws.com" },
```

```
"sa-east-1":{ "EC2Principal":"ec2.amazonaws.com",
  "OpsWorksPrincipal":"opsworks.amazonaws.com" },
"cn-north-1":{ "EC2Principal":"ec2.amazonaws.com.cn",
  "OpsWorksPrincipal":"opsworks.amazonaws.com.cn" },
"eu-central-1":{ "EC2Principal":"ec2.amazonaws.com",
  "OpsWorksPrincipal":"opsworks.amazonaws.com" }
},
"AWSInstanceType2Virt":{
  "t2.micro":{"Virt":"HVM" },
  "t2.small":{"Virt":"HVM"},
  "t2.medium":{"Virt":"HVM"},
  "t2.large":{"Virt":"HVM" }
},
"AWSInstanceType2EBSOpt":{
  "t2.micro":{"EBSOpt":"false"},
  "t2.small":{"EBSOpt":"false"},
  "t2.medium":{"EBSOpt":"false"},
  "t2.large":{"EBSOpt":"false"}
},
"AWSRegionVirt2AMI":{
  "us-east-1":{"HVM":"ami-8fcee4e5"},
  "us-west-2":{"HVM":"ami-63b25203"}
},
"AWSRegionToAMI":{
  "us-east-1":{"AMIID":"ami-2b3b6041"},
  "us-west-2":{"AMIID":"ami-ac6872cd"},
  "eu-west-1":{"AMIID":"ami-03238b70"},
  "ap-northeast-1":{"AMIID":"ami-fb2f1295"},
  "ap-southeast-2":{"AMIID":"ami-43547120"},
  "us-west-1":{"AMIID":"ami-bfe095df"},
  "ap-southeast-1":{"AMIID":"ami-c78f43a4"},
  "eu-central-1":{"AMIID":"ami-e1e6f88d"}}},
```

```
"Resources":{ "CPOpsDeploySecGroup":{
  "Type":"AWS::EC2::SecurityGroup",
  "Properties":{"GroupDescription":"Lets you manage OpsWorks instances deployed to by
CodePipeline"}},
"CPOpsDeploySecGroupIngressHTTP":{
  "Type":"AWS::EC2::SecurityGroupIngress",
  "Properties":{"IpProtocol":"tcp",
    "FromPort":"80",
    "ToPort":"80",
    "CidrIp":"0.0.0.0/0",
    "GroupId":{
      "Fn::GetAtt":["CPOpsDeploySecGroup",
        "GroupId" ]
    }
  }},
"CPOpsDeploySecGroupIngressSSH":{"Type":"AWS::EC2::SecurityGroupIngress",
  "Properties":{"IpProtocol":"tcp",
    "FromPort":"22",
    "ToPort":"22",
    "CidrIp":"0.0.0.0/0",
    "GroupId":{
      "Fn::GetAtt":[
        "CPOpsDeploySecGroup",
        "GroupId" ]
    }
  }},
"MyStack":{
  "Type":"AWS::OpsWorks::Stack",
  "Properties":{
    "Name":{"Ref":"AWS::StackName" },
    "ServiceRoleArn":{
      "Fn::GetAtt":["OpsWorksServiceRole",
        "Arn" ]},
    "ConfigurationManager":{"Name":"Chef",
      "Version":"11.10"
    },
  },
}
```

```
"DefaultOs":"Amazon Linux 2016.03",
  "DefaultInstanceProfileArn":{"Fn::GetAtt":[
    "OpsWorksInstanceProfile",
    "Arn"]
  } } },
"MyLayer":{"Type":"AWS::OpsWorks::Layer",
"Properties":{"StackId":{"Ref":"MyStack"},
  "Name":"MyLayer",
  "Type":"php-app",
  "Shortname":"mylayer",
  "EnableAutoHealing":"true",
  "AutoAssignElasticIps":"false",
  "AutoAssignPublicIps":"true",
  "CustomSecurityGroupIds":[
    { "Fn::GetAtt":[
      "CPOpsDeploySecGroup",
      "GroupId" ]
    }
  ] },
"DependsOn":["MyStack",
  "CPOpsDeploySecGroup" ]},
"OpsWorksServiceRole":{"
  "Type":"AWS::IAM::Role",
  "Properties":{"AssumeRolePolicyDocument":{"
    "Statement":[
      {"Effect":"Allow",
        "Principal":{"
          "Service":[{" Fn::FindInMap":[
            "Region2Principal",
            {"Ref":"AWS::Region"},
            "OpsWorksPrincipal"
          ] } ] },
        "Action":["sts:AssumeRole" ] } ] },
```

```

"Path":"/",
  "Policies":[ {
    "PolicyName":"opsworks-service",
    "PolicyDocument":{
      "Statement":[ { "Effect":"Allow",
        "Action":[ "ec2:*",
          "iam:PassRole",
          "cloudwatch:GetMetricStatistics",
          "elasticloadbalancing:*" ],
        "Resource":"*"
      } ] } ] } ],
  "OpsWorksInstanceProfile":{ "Type":"AWS::IAM::InstanceProfile",
    "Properties":{
      "Path":"/",
      "Roles":[ { "Ref":"OpsWorksInstanceRole" } ] }
    },
    "OpsWorksInstanceRole":{ "Type":"AWS::IAM::Role",
      "Properties":{
        "AssumeRolePolicyDocument":{
          "Statement":[ { "Effect":"Allow",
            "Principal":{
              "Service":[ { "Fn::FindInMap":[
                "Region2Principal",
                { "Ref":"AWS::Region" },
                "EC2Principal"
              ] } ] },
            "Action":["sts:AssumeRole" ]
          } ] },
          "Path":"/",
          "Policies":[ { "PolicyName":"s3-get",
            "PolicyDocument":{ "Version":"2012-10-17",
              "Statement":[ {

```

```

"Effect": "Allow",
    "Action": [ "s3:GetObject" ],
    "Resource": "*"
  } ] } ] } },
"MyInstance": { "Type": "AWS::OpsWorks::Instance",
  "Properties": {
    "LayerIds": [ { "Ref": "MyLayer" } ],
    "StackId": { "Ref": "MyStack" },
    "InstanceType": "c3.medium",
    "SshKeyName": { "Ref": "KeyName" }
  } },
"CodePipelineTrustRole": { "Type": "AWS::IAM::Role",
  "Properties": { "AssumeRolePolicyDocument": {
    "Statement": [
      { "Sid": "1",
        "Effect": "Allow",
        "Principal": { "Service": [ "codepipeline.amazonaws.com" ] },
        "Action": "sts:AssumeRole" }
    ] },
    "Path": "/" } },
"CodePipelineRolePolicies": {
  "Type": "AWS::IAM::Policy",
  "Properties": {
    "PolicyName": "CodePipelinePolicy",
    "PolicyDocument": {
      "Statement": [ { "Action": [ "s3:*" ],
        "Resource": [ "*" ],
        "Effect": "Allow" },
        { "Action": [
          "s3:PutBucketPolicy",
          "s3:PutObject" ],
          "Resource": [ {
            "Fn::Join": [ "",
              [

```



```
[ "arn:aws:s3::demo-",
  {"Ref":"AWS::Region" },
    "-",
    {"Ref":"AWS::AccountId" },
    "-",
    { "Ref":"AWS::StackName" }
  ] ] },
  "Effect":"Allow" },
  { "Action":["codecommit:GetBranch",
    "codecommit:GetCommit",
    "codecommit:UploadArchive",
    "codecommit:GetUploadArchiveStatus",
    "codecommit:CancelUploadArchive" ],
    "Resource":"*",
    "Effect":"Allow"},
  {"Action":["codepipeline:*",
    "iam:ListRoles",
    "iam:PassRole",
    "opsworks:*",
    "elasticbeanstalk:DescribeApplications",
    "elasticbeanstalk:DescribeEnvironments",
    "lambda:GetFunctionConfiguration",
    "lambda:ListFunctions" ],
    "Resource":"*",
    "Effect":"Allow"
  } ] },
  "Roles":[
    { "Ref":"CodePipelineTrustRole" }
  ] } },
  "MyOpsWorksApp":{
    "Type":"AWS::OpsWorks::App",
    "Properties":{"StackId":{"Ref":"MyStack" },
      "Type":"php",
      "Shortname":"phptestapp",
      "Name":{
        "Ref":"AWS::StackName"
      },
      "AppSource":{
        "Type":"other"
      }
    }
  }
}
```

```

"Shortname":"phptestapp",
  "Name":{"Ref":"AWS::StackName" },
  "AppSource":{
    "Type":"other"
  }},
"MyPipeline":{"Type":"AWS::CodePipeline::Pipeline",
"DependsOn":["MyStack",
  "MyOpsWorksApp",
  "MyLayer",
  "MyInstance"],
"Properties":{
  "Name":{
    "Fn::Join":["",
      [{ "Ref":"AWS::StackName"},
        "-Pipeline"] ] },
  "RoleArn":{
    "Fn::GetAtt":[
      "CodePipelineTrustRole",
      "Arn" ] },
  "Stages":[
    { "Name":"Source",
      "Actions":[{"InputArtifacts":[ ],
        "Name":"Source",
        "ActionTypeId":{
          "Category":"Source",
          "Owner":"AWS",
          "Version":"1",
          "Provider":"CodeCommit"},
        "OutputArtifacts":[{"Name":"MyApp" } ],
        "Configuration":{"BranchName":{"Ref":"RepositoryBranch" },
          "RepositoryName":{"Ref":"RepositoryName" } },
        "RunOrder":1
      } ] },
    } ] },

```

```

{ "Name": "Deploy",
  "Actions": [ { "InputArtifacts": [ { "Name": "MyApp" } ],
    "Name": "DeployPHPApp",
    "ActionTypeId": { "Category": "Deploy",
      "Owner": "AWS",
      "Version": "1",
      "Provider": "OpsWorks" },
    "OutputArtifacts": [ ],
    "Configuration": {
      "StackId": { "Ref": "MyStack" },
      "AppId": { "Ref": "MyOpsWorksApp" },
      "DeploymentType": "deploy_app",
      "LayerId": { "Ref": "MyLayer" } },
    "RunOrder": 1
  } ] },
  "ArtifactStore": {
    "Type": "S3",
    "Location": { "Fn::Join": [ "",
      [ "codepipeline-",
        { "Ref": "AWS::Region" },
        "-",
        { "Ref": "AWS::AccountId" } ] ] } },
    "Name": "codepipeline-artifacts",
    "EncryptionKey": { "Ref": "AWS::KMS::Key" } } },
  "Outputs": { "PipelineUrl": {
    "Value": {
      "Fn::Join": [ "",
        [ "https://console.aws.amazon.com/codepipeline/home?region=",
          { "Ref": "AWS::Region" },
          "#/view/",
          { "Ref": "MyPipeline" } ] ] },
    "Description": "Pipeline Url" } } }

```

- Upload the code to CloudFormation in Virginia region and click on next
- Give a name for your stack
- Choose a key name you have in that region

Stack name: OpsWorks-codepipe

Parameters

Dynamic Configuration

KeyName: ishu-virginia  
Name of an existing EC2 KeyPair for SSH access to the instances.

CodeCommit Configuration

RepositoryName: opsworks-php-demo  
The name of the CodeCommit repo

RepositoryBranch: master  
The name of the branch for the CodeCommit repo

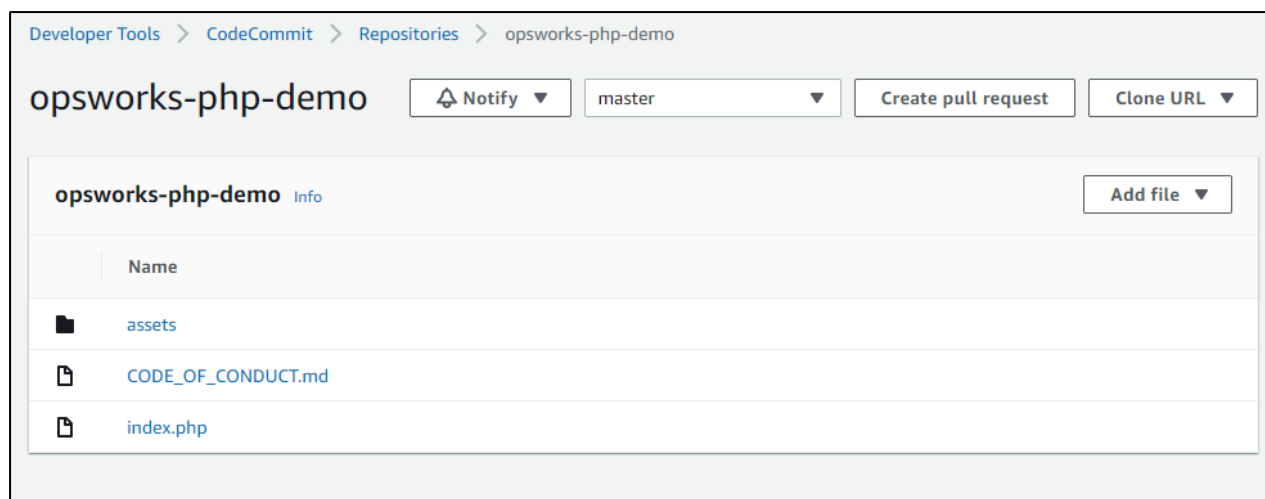
Cancel Previous **Next**

- Click on next
- Select the Check box to give permission to your CloudFormation to create IAM role
- Then, Click On create

**Note:** Check whether if you have a S3 in the name of “codepipeline-us-east-1-  
<accountNumber>”, if not create it

### Step 2: Create a CodeCommit Repository

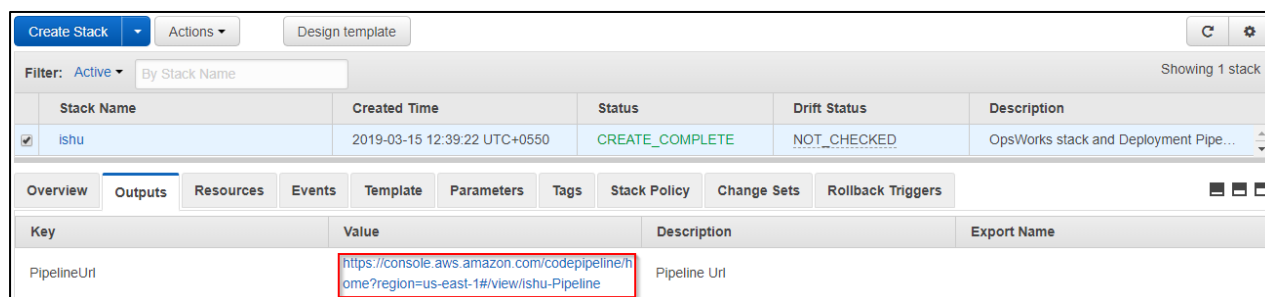
- In your console search for CodeCommit and select it
- Create a repository in the name of “opsworks-php-demo”
- Download the application files <https://s3.amazonaws.com/opsworks-codepipeline-demo/opsworks-nodejs-demo-app.zip>
- Push the files to your repository



**Note:** Make sure your index and the assests file are in the root of the repository

### Step 3: Test your application

- Click on the link which is there as the output from cloudFormation



- Click on the Stack

View progress and manage your pipeline.

[Edit](#) [Release change](#)

### Source

Source ⓘ

[AWS CodeCommit](#)

✓ Succeeded 5 min ago  
ec9c4aa

Source: Added bootstrap....

### Deploy

DeployPHPApp ⓘ

[AWS OpsWorks Stacks](#)

✓ Succeeded 4 min ago  
[Details](#)

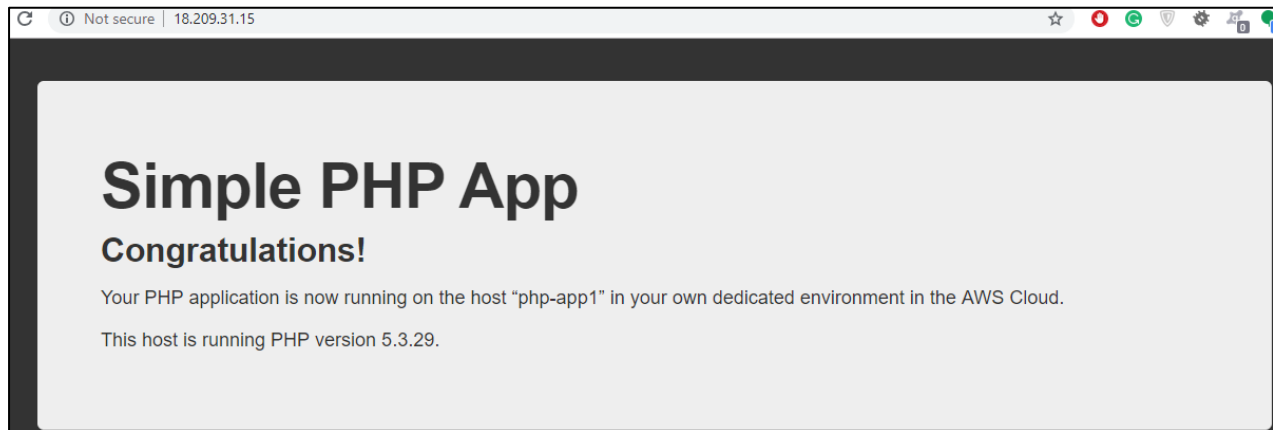
Source: Added bootstrap....

- In that Select Instance on the navigation bar and click on the IP address to test your application

MyLayer

Search for instances in this layer by name, status, size, type, AZ or IP

Hostname	Status	Size	Type	AZ	Public IP	Actions
php-app1	online	c3.large	24/7	us-east-1a	18.209.31.15	stop ssh
+ Instance						



- If you Update the Code in the code repository it gets automatically replicated in the stack

