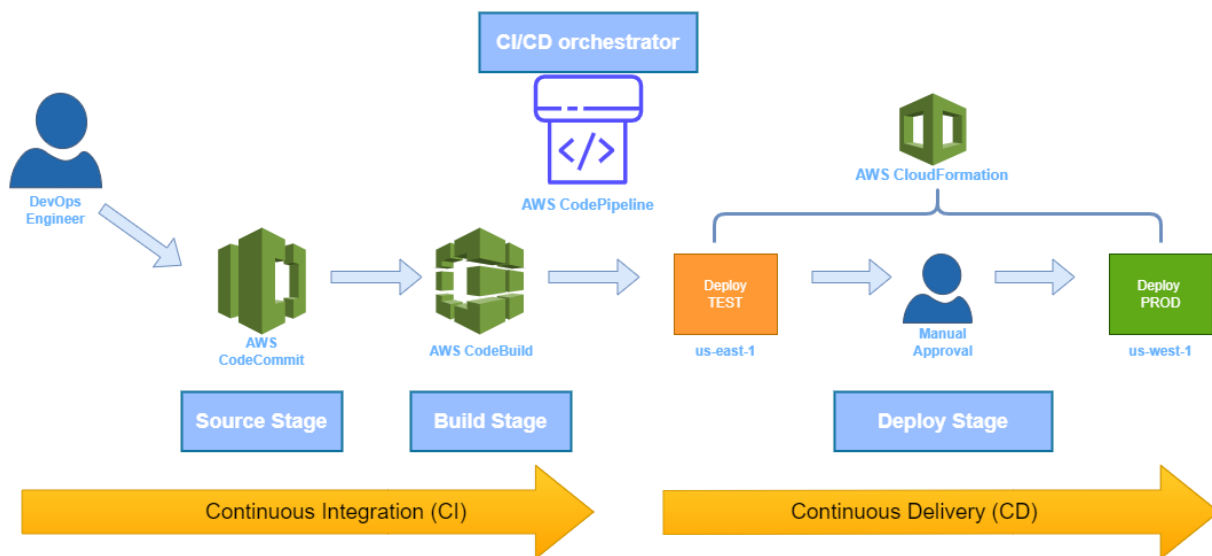


Building a four-stage CI/CD pipeline for multi-region deployment with AWS CodePipeline

Variant 1: Source Stage - AWS CodeCommit

CI/CD Pipeline Architecture



- Launch Amazon Linux 2 instance (Code Build supports Amazon Linux 2 and ubuntu and we want our testing environment to be as similar as possible to our automated environment) with full administrative access and use it as CLI to manually test the commands for the buildspec. Take the CloudFormation template from an s3 bucket called bcktr (in the pipeline the template will come from the Source stage).

```
[ec2-user@ip-10-0-1-43 ~]$ aws s3 ls
2022-02-01 05:59:00 aws-cloudtrail-logs-399866859445-61abd601
2021-10-27 20:32:10 bcktr
2022-03-08 23:02:34 cf-templates-18tpur7rrqhd-y-us-east-1
2022-03-08 23:31:42 cf-templates-18tpur7rrqhd-y-us-west-1
2022-03-06 06:58:33 codepipeline-us-east-1-289786467232
2022-03-06 06:40:17 party-website-bckt
[ec2-user@ip-10-0-1-43 ~]$ aws s3 ls s3://bcktr
2021-11-15 03:04:48      37 cf.html
2021-11-15 02:29:27       5 error.html
2022-03-09 02:20:38   2178 webserver-template.yml
[ec2-user@ip-10-0-1-43 ~]$
```

```
[ec2-user@ip-10-0-1-43 ~]$ aws s3 cp s3://bcktr/webserver-template.yml .
download: s3://bcktr/webserver-template.yml to ./webserver-template.yml
[ec2-user@ip-10-0-1-43 ~]$ ls -l
total 4
-rw-rw-r-- 1 ec2-user ec2-user 2178 Mar  9 02:20 webserver-template.yml
[ec2-user@ip-10-0-1-43 ~]$
```

```

  _l  _l_ )
 _l  (  /
 _l\_\_l_

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 5 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-1-43 ~]$ aws cloudformation package --template-file webserver-template.yaml --s3-bucket bcktr --output-template-file outputTemplate.yaml

Invalid template path webserver-template.yaml
[ec2-user@ip-10-0-1-43 ~]$ aws cloudformation package --template-file webserver-template.yaml --s3-bucket bcktr --output-template-file outputTemplate.yaml

Invalid template path webserver-template.yaml
[ec2-user@ip-10-0-1-43 ~]$ ls -l
total 4
-rw-rw-r-- 1 ec2-user ec2-user 2178 Mar  9 02:20 webserver-template.yaml
[ec2-user@ip-10-0-1-43 ~]$ mv webserver-template.yaml webserver-template.yml
[ec2-user@ip-10-0-1-43 ~]$ ls -l
total 4
-rw-rw-r-- 1 ec2-user ec2-user 2178 Mar  9 02:20 webserver-template.yml
[ec2-user@ip-10-0-1-43 ~]$ aws cloudformation package --template-file webserver-template.yml --s3-bucket bcktr --output-template-file outputTemplate.yaml

Successfully packaged artifacts and wrote output template to file outputTemplate.yaml.
Execute the following command to deploy the packaged template
aws cloudformation deploy --template-file /home/ec2-user/outputTemplate.yaml --stack-name <YOUR STACK NAME>
[ec2-user@ip-10-0-1-43 ~]$
```

- In VS Code create:
- the CloudFormation **template** for the webserver

```
3-stage-pipeline > 3-4-stage-pipeline-repo > ! webserver-template.yaml

1 Parameters:
2   InstanceType:
3     Description: EC2 instance type.
4     Type: String
5     Default: t2.micro
6   KeyName:
7     Description: Name of an existing EC2 key pair for SSH access to the EC2 instance.
8     Type: AWS::EC2::KeyPair::KeyName
9   SSHLocation:
10    Description: The IP address range that can be used to SSH to the EC2 instances
11    Type: String
12    MinLength: '9'
13    MaxLength: '18'
14    AllowedPattern: "(\d{1,3})\.\.(\d{1,3})\.\.(\d{1,3})\.\.(\d{1,3})/(\d{1,2})" # IP Address
15    ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x.
16    Default: 0.0.0.0/0
17 SubnetId:
18   Type: AWS::EC2::Subnet::Id
19   Description: Choose subnet to launch the server in
20 VpcId:
21   Type: AWS::EC2::VPC::Id
22 ImageId:
23   Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
24   Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
25
```

```

26 Resources:
27   WebServer:
28     Type: AWS::EC2::Instance
29     Properties:
30       ImageId: !Ref ImageId
31       InstanceType: !Ref InstanceType # Nothing is hardcoded
32       SubnetId: !Ref SubnetId
33       KeyName: !Ref KeyName
34       SecurityGroupIds:
35         - !Ref WebServerSecurityGroup
36       UserData:
37         Fn::Base64: !Sub |
38           #!/bin/bash
39           yum update -y
40           yum install -y httpd
41           systemctl start httpd
42           systemctl enable httpd
43           echo "<html><body><h1>Hello from Server in ${AWS::Region}</body></html>" > /var/www/html/index.html
44
45   WebServerSecurityGroup: #security group ID
46     Type: AWS::EC2::SecurityGroup
47     Properties:
48       GroupDescription: 'Enable HTTP access via port 80 + SSH access'
49       SecurityGroupIngress:
50         - CidrIp: 0.0.0.0/0
51           FromPort: '80'
52           IpProtocol: tcp
53           ToPort: '80'
54         - CidrIp: !Ref SSHLocation
55           FromPort: '22'
56           IpProtocol: tcp
57           ToPort: '22'
58       VpcId: !Ref VpcId
59
60   Outputs:
61     InstanceId:
62       Description: InstanceId of the newly created EC2 instance
63       Value: !GetAtt WebServer.PublicIp
64     DnsName:
65       Description: DNS name of the newly created EC2 instance
66       Value: !GetAtt WebServer.PublicDnsName

```

- the cheat sheet and put the manually tested commands /use it to make the buildspec/

```

3-stage-pipeline > $ cheatsheet
1  TESTING STEPS:
2
3  # Check AWS CLI installed
4  aws --version
5
6  # Validating that Python is installed
7  python --version
8
9  # Validating that the PYTHON PACKAGE MANAGER - pip3 is installed
10 pip3 --version
11
12 # Installing CFN-LINT - https://github.com/stepanova13/cfn-lint/blob/main/README.md
13 pip3 install cfn-lint
14
15 # Checking cfn version
16 cfn-lint --version
17
18 # Validating the cfn template
19 cfn-lint network-template.yaml
20
21 # Building our cfn template - https://docs.aws.amazon.com/cli/latest/reference/cloudformation/package.html
22 aws cloudformation package --template-file webserver-template.yaml --s3-bucket bcktr --output-template-file outputTemplate.yaml
23
24

```

- the **buildspec** with the instructions for the Build stage. Automatically tests the CloudFormation template for the webserver / created with the commands from the cheat sheet/. After testing the template creates a package with the webserver-template.yaml, puts it in s3 bucket called bcktr and in a file called outputTemplate.yaml - <https://docs.aws.amazon.com/cli/latest/reference/cloudformation/package.html> .Then creates an artifact with the 2 files - outputTemplate.yaml and webserver-template.yaml zipped together and sends the artifact to the next stage – Deploy.

Examples

Following command exports a template named template.json by uploading local artifacts to S3 bucket bucket-name and writes the exported template to packaged-template.json:

```
aws cloudformation package --template-file /path_to_template/template.json --s3-bucket bucket-name --output-template-file packaged-template.json
```

```

3-stage-pipeline > 3-4-stage-pipeline-repo > ! buildspec.yaml
1  version: 0.2
2
3  phases:
4    install:
5      runtime-versions:
6        python: 3.8
7    pre_build:
8      commands:
9        - echo "Check if AWS CLI is installed.."
10       - aws --version
11       - echo "Validating that Python is installed.."
12       - python --version
13       - echo "Validating that the PYTHON PACKAGE MANAGER - pip3 is installed.."
14       - pip3 --version
15       - echo Installing CFN-LINT..
16       - pip3 install cfn-lint
17       - echo Checking CFN version..
18       - cfn-lint --version
19    build:
20      commands:
21        - echo Validating the cfn template
22        - cfn-lint webserver-template.yaml
23        - echo cfn template build started on `date`
24        - aws cloudformation package --template-file webserver-template.yaml --s3-bucket bcktr --output-template-file outputTemplate.yaml
25    post_build:
26      commands:
27        - echo Build completed on `date`
28  artifacts:
29    files:
30      - outputTemplate.yaml # zipped in our build artifact
31      - webserver-template.yaml
32

```

- Create **repository** in Code Commit and use it at the Source stage

The screenshot shows the AWS CodeCommit console interface. On the left, the 'Developer Tools' sidebar is open, showing the 'CodeCommit' section with a list of actions: 'Source', 'Getting started', 'Repositories', 'Code' (highlighted), 'Pull requests', 'Commits', 'Branches', 'Git tags', and 'Settings'. The main content area displays the details for the '3-4-stage-pipeline-repo' repository. At the top, there are navigation breadcrumbs: 'Developer Tools > CodeCommit > Repositories > 3-4-stage-pipeline-repo'. Below this, the repository name '3-4-stage-pipeline-repo' is shown with a 'Notify' button and a 'master' branch selector. A table lists the repository's files:

| Name |
|-------------------------|
| buildspec.yaml |
| webserver-template.yaml |

- Build **CI/CD pipeline** with CodePipeline

3-4-stage-pipeline

- Notify ▼
- Edit
- Stop execution
- Clone pipeline
- Release change

✔ Source Succeeded

Pipeline execution ID: 30425dcc-3f98-491c-8bd3-ca39a9cf8531

Source ⓘ

AWS CodeCommit

✔ Succeeded - 10 minutes ago

8014bce2

8014bce2 Source: changed TheVpcId --> VpcId & TheSubnetId --> SubnetId

- ✔
- ✔
- ✔



Disable transition

✔ Build Succeeded

Pipeline execution ID: 30425dcc-3f98-491c-8bd3-ca39a9cf8531

Build ⓘ

AWS CodeBuild

✔ Succeeded - 11 minutes ago

Details

8014bce2 Source: changed TheVpcId --> VpcId & TheSubnetId --> SubnetId

- ✔
- ✔
- ✔



Disable transition

Deploy

Succeeded

Pipeline execution ID: 30425dcc-3f98-491c-8bd3-ca39a9cf8531

TEST

AWS CloudFormation

Succeeded

- 9 minutes ago

Details

↓

Approval

Manual approval

Approved

- 8 minutes ago

Details

↓

PROD

AWS CloudFormation

Succeeded

- 7 minutes ago

Details

8014bce2 Source: changed TheVpcId --> VpcId & TheSubnetId --> SubnetId

✓

✓

✓

- The CloudFormation stack created by the pipeline in N. Virginia region

test-stack

Delete

Update

Stack actions

Create stack

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Parameters (6)

Search parameters

| Key | Value | Resolved value |
|--------------|---|-----------------------|
| ImageId | /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 | ami-01b20f5ea962e3fe7 |
| InstanceType | t2.micro | - |
| KeyName | new_key | - |
| SSHLocation | 0.0.0.0/0 | - |
| SubnetId | subnet-071fe815cd6902f15 | - |
| VpcId | vpc-09c6bf7babd83473f | - |

- passed Parameter overrides in Code Pipeline in N. Virginia region

▼ Advanced

Parameter overrides

```
{ "KeyName": "new_key", "SubnetId": "subnet-071fe815cd6902f15", "VpcId": "vpc-09c6bf7babb83473f" }
```

{ "parameterName": "value" }

Variable namespace - *optional*

Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

Output artifacts

Choose a name for the output of this action.


No more than 100 characters



- The CloudFormation stack created by the pipeline in N. California region

prod-stack

Delete Update Stack actions ▼ Create stack ▼

Stack info Events Resources Outputs Parameters Template Change sets

Parameters (6) 

 Search parameters 

| Key ▲ | Value ▼ | Resolved value ▼ |
|--------------|---|-----------------------|
| ImageId | /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 | ami-0577aef94c154720e |
| InstanceType | t2.micro | - |
| KeyName | ca-key-pair | - |
| SSHLocation | 0.0.0.0/0 | - |
| SubnetId | subnet-04133be9525e40f50 | - |
| VpcId | vpc-0855650485c0c29fe | - |

- passed Parameter overrides in Code Pipeline in N. California region

▼ Advanced

Parameter overrides

```
{ "KeyName": "ca-key-pair", "SubnetId": "subnet-04133be9525e40f50", "VpcId": "vpc-0855650485c0c29fe" }
```

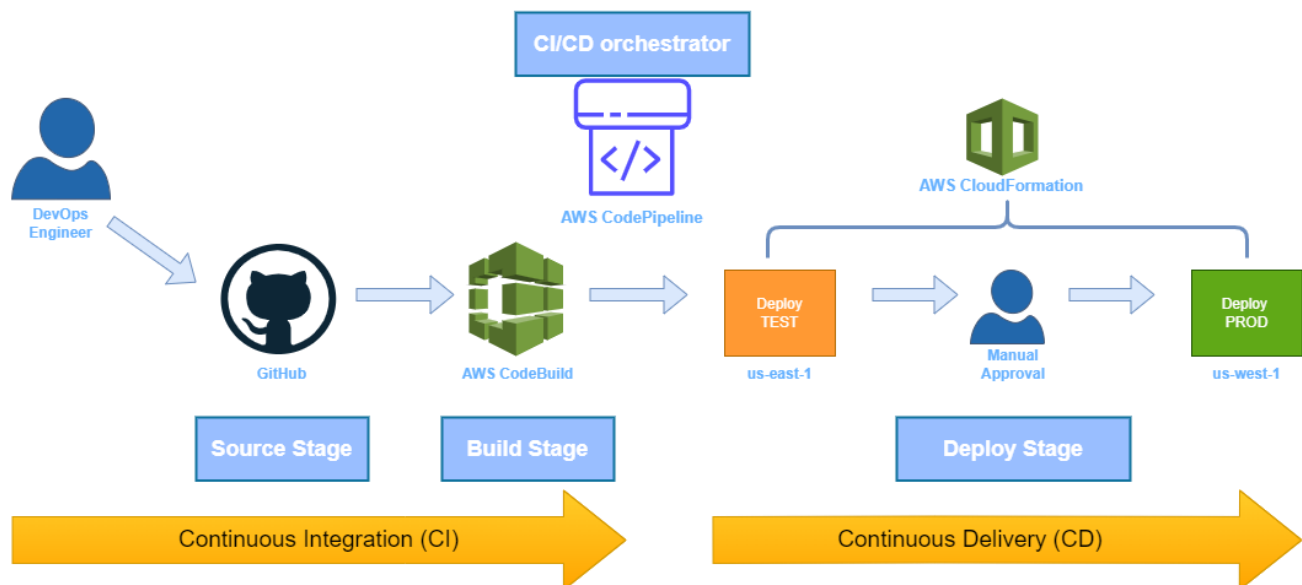
```
{ "parameterName": "value" }
```

Variable namespace - *optional*
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

Output artifacts
Choose a name for the output of this action.

Variant 2: Source Stage - GitHub

CI/CD Pipeline Architecture



3-stage-pipeline

Notify ▼

Edit

Stop execution

Clone pipeline


Release change

✔ Source


Succeeded


Pipeline execution ID: 7e01c0ed-88c9-4ef7-aafb-0d315d2c7741

Source

GitHub (Version 1) 

✔ Succeeded - 3 minutes ago

370f1f6d 

370f1f6d  Source: webserver-template.yaml changed

✔

✔

✔

↓


Disable transition

✔ Build


Succeeded


Pipeline execution ID: 7e01c0ed-88c9-4ef7-aafb-0d315d2c7741

Build

AWS CodeBuild 

✔ Succeeded - 3 minutes ago

Details 

370f1f6d  Source: webserver-template.yaml changed

✔

✔

✔

↓


Disable transition

✔ Deploy


Succeeded

Pipeline execution ID: 7e01c0ed-88c9-4ef7-aafb-0d315d2c7741

Deploy

AWS CloudFormation 

✔ Succeeded - 3 minutes ago


Details 

↓

Manual-Approval


Manual approval

✔ Approved - 2 minutes ago


Details 


↓

Delivery

AWS CloudFormation 

✔ Succeeded - 1 minute ago

Details 

370f1f6d  Source: webserver-template.yaml changed

✔

✔

✔