

Cloudformation

1. Start a EC2 Instance with Parameter KEY NAME, VPC, Subnet using Cloudformation Template.

```
index.html x vpc-igw-subnet-routetable-route-sg-ec2.yaml x
1 Resources:
2   MyVPC:
3     Type: AWS::EC2::VPC
4     Properties:
5       CidrBlock: 10.0.0.0/16
6       EnableDnsSupport: true
7       EnableDnsHostnames: true
8       Tags:
9         - Key: owner
10           Value: Jay
11         - Key: purpose
12           Value: Bootcamp
13   IGW:
14     Type: AWS::EC2::InternetGateway
15     Properties:
16       Tags:
17         - Key: owner
18           Value: jay
19         - Key: purpose
20           Value: bootcamp
21   IGWAttachment:
22     Type: AWS::EC2::VPCEGatewayAttachment
23     Properties:
24       InternetGatewayId:
25         Ref: IGW
26       VpcId:
```

```

27     Ref: MyVPC
28   PublicSubnet1:
29     Type: AWS::EC2::Subnet
30     Properties:
31       VpcId:
32         Ref: MyVPC
33       AvailabilityZone: "us-east-1a"
34       CidrBlock: 10.0.1.0/24
35       MapPublicIpOnLaunch: true
36       Tags:
37         - Key: owner
38           Value: jay
39         - Key: purpose
40           Value: bootcamp
41   PublicSubnet2:
42     Type: AWS::EC2::Subnet
43     Properties:
44       VpcId:
45         Ref: MyVPC
46       AvailabilityZone: "us-east-1b"
47       CidrBlock: 10.0.2.0/24
48       MapPublicIpOnLaunch: true
49       Tags:
50         - Key: owner

```

```

    Value: bootcamp
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId:
      Ref: MyVPC
PublicRoute:
  Type: AWS::EC2::Route
  DependsOn: IGWAttachment # If IGW is attached with VPC, then only
  Properties:
    RouteTableId:
      Ref: PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId:
      Ref: IGW
PublicSubnet1Association:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId:
      Ref: PublicRouteTable
    SubnetId:
      Ref: PublicSubnet1

```

```

    Ref: PublicSubnet1
PublicSubnet2Association:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId:
      Ref: PublicRouteTable
    SubnetId:
      Ref: PublicSubnet2
SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: "Test Gecurity Group"
    VpcId:
      Ref: MyVPC
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        FromPort: 22
        ToPort: 22
        IpProtocol: tcp
      - CidrIp: 0.0.0.0/0
        FromPort: 80
        ToPort: 80
        IpProtocol: tcp

```

```

    IpProtocol: tcp
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: "ami-085925f297f89fce1"
    InstanceType: "t2.micro"
    KeyName: Jay
    #VpcId: # Not Supported Here
    # Ref: MyVPC
    SubnetId:
      Ref: PublicSubnet1
    SecurityGroupIds:
      - Ref: SecurityGroup

```

CloudFormation > Stacks > vpc-ec2

Stacks (1)

Filter by stack name

Active ☐ View nested ☒

< 1 >

vpc-ec2

2020-05-11 13:25:09 UTC+0530

CREATE_COMPLETE

Stack info | Events | **Resources** | Outputs | Parameters | Template | Change sets

Resources (11)

Search resources

Logical ID	Physical ID	Type	Status
EC2Instance	i-01e582543c459cf7e	AWS::EC2::Instance	CREATE_COMPLETE
IGW	igw-0ea808e18f7cfbaec	AWS::EC2::InternetGateway	CREATE_COMPLETE
IGWAttachment	vpc-e-IGWAt-MZ8IDLXZO0V6	AWS::EC2::VPCGatewayAttachment	CREATE_COMPLETE
MyVPC	vpc-034a4cb547518eef4	AWS::EC2::VPC	CREATE_COMPLETE

Launch Instance

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
<input checked="" type="checkbox"/>	i-01e582543c459cf7e	t2.micro	us-east-1a	running	Initializing
<input type="checkbox"/>	i-093fdeb434a448cde	t2.micro	us-east-1c	stopped	

Description | Status Checks | Monitoring | **Tags**

Add/Edit Tags

Key	Value
aws:cloudformation:logical-id	EC2Instance
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:315002452909:stack/vpc-ec2/bc4da1a0-935c-11ea-8b3d-0a8267cebc39
aws:cloudformation:stack-name	vpc-ec2

2. Instll Nginx in EC2 and put it in a ASG.

i) First by Userdata

ii) By Metadata

By userdata

```
Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: "Test-Template"
      LaunchTemplateData:
        InstanceType: t2.micro
        KeyName: Jay
        ImageId: ami-04b2519c83e2a7ea5
        UserData:
          Fn::Base64:
            Fn::Sub: |
              #!/bin/bash
              apt-get update -y
              apt-get install nginx -y
              systemctl restart nginx
  MyASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      LaunchTemplate:
        LaunchTemplateId:
          Ref: MyLaunchTemplate
        Version:
          # If we will not put "Fn" then it is throwing error "Version must be a string"
          Fn::GetAtt: "MyLaunchTemplate.LatestVersionNumber"
      MinSize: 1
```

```
MyASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId:
        Ref: MyLaunchTemplate
      Version:
        # If we will not put "Fn" then it is throwing error "Version must be a string" t
        Fn::GetAtt: "MyLaunchTemplate.LatestVersionNumber"
    MinSize: 1
    MaxSize: 2
    AvailabilityZones:
      Fn::GetAZs: ""
```

Timestamp	Logical ID	Status	Status reason
2020-05-13 00:57:53 UTC+0530	test1	CREATE_COMPLETE	-
2020-05-13 00:57:52 UTC+0530	MyASG	CREATE_COMPLETE	-
2020-05-13 00:56:35 UTC+0530	MyASG	CREATE_IN_PROGRESS	Resource creation Initiated
2020-05-13 00:56:33 UTC+0530	MyASG	CREATE_IN_PROGRESS	-

i-Od837069bb337e8... t2.micro ap-south-1a running Initializing None ec2-13-126-168-26.a

Instance: **i-Od837069bb337e8e2** Public DNS: **ec2-13-126-168-26.ap-south-1.compute.amazonaws.com**

Description Status Checks Monitoring **Tags**

Add/Edit Tags

Key	Value	
aws:autoscaling:groupName	test1-MyASG-1DGGWB4EMAC22	Show Column
aws:cloudformation:logical-id	MyASG	Show Column
aws:cloudformation:stack-id	arn:aws:cloudformation:ap-south-1:315002452909:stack/test1/798884e0-9486-11ea-9c79-0aa78e4848ba	Show Column
aws:cloudformation:stack-name	test1	Show Column
aws:ec2:launchtemplate:id	lt-041748c0c283e9dfd	Show Column

View/Change User Data

Instance ID: **i-Od837069bb337e8e2**

User Data:

```
#!/bin/bash
apt-get update -y
apt-get install nginx -y
systemctl restart nginx
```

To edit your instance's user data you first need to stop your instance.

Cancel Save

Create Auto Scaling group

Actions

Filter:

1 to 1 of 1 Auto Scaling Groups

<input type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check
<input type="checkbox"/>	test1-MyASG-...	Test-Template	1	1	1	2	ap-south-1c, ap-south-1b, a...	300	0

By metadata

```
Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            yum:
              nginx: []
    Properties:
      LaunchTemplateName: "Test-Template"
      LaunchTemplateData:
        InstanceType: t2.micro
        KeyName: Jay
        ImageId: ami-04b2519c83e2a7ea5
        SecurityGroupIds:
          - sg-057ad8e5c2a83dd43
        UserData:
          Fn::Base64:
            Fn::Sub: |
              #!/bin/bash
              yum update -y aws-cfn-bootstrap
              service nginx start
              /opt/aws/bin/cfn-init -s ${AWS::StackId} -r MyLaunchTemplate --region ${AWS::Region}
              /opt/aws/bin/cfn-hup || error_exit "Failed to start cfn-hup"
              /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackId} --resource MyLaunchTemplate
```

```

- sg-057ad8e5c2a83dd43
UserData:
  Fn::Base64:
    Fn::Sub: |
      #!/bin/bash
      yum update -y aws-cfn-bootstrap
      service nginx start
      /opt/aws/bin/cfn-init -s ${AWS::StackId} -r MyLaunchTemplate --region ${AWS:
      /opt/aws/bin/cfn-hup || error_exit "Failed to start cfn-hup"
      /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackId} --resource MyLaunchTemp
MyASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId:
        Ref: MyLaunchTemplate
      Version:
        Fn::GetAtt: "MyLaunchTemplate.LatestVersionNumber"
    MinSize: 1
    MaxSize: 2
    AvailabilityZones:
      Fn::GetAZs: "" # Getting all the availability zones

```

Stacks (1)

Filter by stack name

Active View nested

test3

2020-05-13 01:06:33 UTC+0530

CREATE_COMPLETE

Events (8)

Search events

Timestamp	Logical ID	Status	Status reason
2020-05-13 01:07:35 UTC+0530	test3	CREATE_COMPLETE	-
2020-05-13 01:07:34 UTC+0530	MyASG	CREATE_COMPLETE	-
2020-05-13 01:06:41 UTC+0530	MyASG	CREATE_IN_PROGRESS	Resource creation Initiated
2020-05-13 01:06:39 UTC+0530	MyASG	CREATE_IN_PROGRESS	-

View/Change User Data



Instance ID: i-0db109dd0af92729f

User Data:

```
#!/bin/bash
yum update -y aws-cfn-bootstrap
service nginx start
/opt/aws/bin/cfn-init -s arn:aws:cloudformation:ap-south-1:315002452909:stack/test3
/e2d18860-9487-11ea-99a3-028f38a10716 -r MyLaunchTemplate --region ap-south-1 ||
error_exit 'Failed to run cfn-init'
/opt/aws/bin/cfn-hup || error_exit "Failed to start cfn-hup"
/opt/aws/bin/cfn-signal -e $? --stack arn:aws:cloudformation:ap-south-
1:315002452909:stack/test3/e2d18860-9487-11ea-99a3-028f38a10716 --resource
MyLaunchTemplate --region ap-south-1
```

To edit your instance's user data you first need to stop your instance.

Filter:

1 to 1 of 1 Auto Scaling Groups

<input checked="" type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check
<input checked="" type="checkbox"/>	test3-MyASG-...	Test-Template	1	1	1	2	ap-south-1c, ap-south-1b, a...	300	0

Auto Scaling Group: test3-MyASG-1J8TPNMWWK8U9

Details

Activity History

Scaling Policies

Instances

Monitoring

Notifications

Tags

Scheduled Actions

Lifecycle Hooks

Add/Edit tags

Key	Value	Tag New Instances
aws:cloudformation:logical-id	MyASG	Yes
aws:cloudformation:stack-id	arn:aws:cloudformation:ap-south-1:315002452909:stack/test3/e2d18860-9487-11ea-99a3-028f38a10716	Yes
aws:cloudformation:stack-name	test3	Yes

3. Create a Sample Index file and copy this file using MetaData into EC2 Instance

```
Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            yum:
              nginx: []
          files:
            "/var/www/html/index.html":
              content:
                Sub: |
                  <br><br>
                  <center>This is <h1>Custom File</h1></center>
              mode: '000400'
              owner: root
              group: root
    Properties:
      LaunchTemplateName: "Test-Template"
      LaunchTemplateData:
        InstanceType: t2.micro
        KeyName: Jay
        ImageId: ami-04b2519c83e2a7ea5
        SecurityGroupIds:
```

```

SecurityGroupIds:
  - sg-057ad8e5c2a83dd43
UserData:
  Fn::Base64:
    Fn::Sub: |
      #!/bin/bash
      yum update -y aws-cfn-bootstrap
      service nginx start
      /opt/aws/bin/cfn-init -s ${AWS::StackId} -r MyLaunchTemplate --r
MyASG:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId:
        Ref: MyLaunchTemplate
      Version:
        Fn::GetAtt: "MyLaunchTemplate.LatestVersionNumber"
    MinSize: 1
    MaxSize: 2
    AvailabilityZones:
      Fn::GetAZs: ""

```

```

[ec2-user@ip-172-31-33-121 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-33-121 html]$ ll
total 4
-r----- 1 root root 68 May 12 19:20 index.html
[ec2-user@ip-172-31-33-121 html]$ cat index.html
cat: index.html: Permission denied
[ec2-user@ip-172-31-33-121 html]$ sudo cat index.html
{"Sub": "<br><br>\n<center>This is <h1>Custom File</h1></center>\n"}[ec2-user@ip-172-31-33-121 html]$ █

```

4. Changing the content of Index should reload the nginx config automatically in EC2 Instance

```

Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Metadata:
      AWS::CloudFormation::Init:
        configSets:
          SetupEnvironment:
            - setupCfnHup
          UpdateEnvironment:
            - updateHostsFile
        setupCfnHup:
          packages:
            yum:
              nginx: []
          files:
            "/var/www/html/index.html":
              content: |
                <br><br>
                <center>This is <h1>Changed File</h1></center>
              mode: '000755'
              owner: ec2-user
              group: ec2-user
            "/etc/cfn/cfn-hup.conf":
              content:
                Fn::Sub: |
                  [main]
                  stack=${AWS::StackId}
                  region=${AWS::Region}
                  interval=1
              mode: '000400'
              owner: root
              group: root
            "/etc/cfn/hooks.d/cfn-auto-reloader.conf":
              content:

```

```

              Fn::Sub: |
                [cfn-auto-reloader-hook]
                triggers=post.update
                path=Resources.MyLaunchTemplate.Metadata.AWS::CloudFormation::Init
                action=/opt/aws/bin/cfn-init --verbose --stack=${AWS::StackName} --region=${AWS::Region} --resource
                runas=root
              mode: '000400'
              owner: root
              group: root
        services:
          sysvinit:
            cfn-hup:
              enabled: true
              ensureRunning: true
              files:
                - '/etc/cfn/cfn-hup.conf'
                - '/etc/cfn/hooks.d/cfn-auto-reloader.conf'
        updateHostsFile :
          commands:
            reload_nginx:
              command: service nginx reload
  Properties:
    LaunchTemplateName: "Test-Template"
    LaunchTemplateData:
      InstanceType: t2.micro
      KeyName: Jay
      ImageId: ami-04b2519c83e2a7ea5
      SecurityGroupIds:
        - sg-057ad8e5c2a83dd43
      UserData:
        Fn::Base64:
          Fn::Sub: |
            #!/bin/bash

```

```

Properties:
  LaunchTemplateName: "Test-Template"
  LaunchTemplateData:
    InstanceType: t2.micro
    KeyName: Jay
    ImageId: ami-04b2519c83e2a7ea5
    SecurityGroupIds:
      - sg-057ad8e5c2a83dd43
    UserData:
      Fn::Base64:
        Fn::Sub: |
          #!/bin/bash
          yum update -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackName} --region ${AWS::Region} --resource MyLaunchTemplate
          rm /usr/share/nginx/html/index.html
          cp /var/www/html/index.html /usr/share/nginx/html/
          service nginx start
yASG:
Type: AWS::AutoScaling::AutoScalingGroup
Properties:
  LaunchTemplate:
    LaunchTemplateId:
      Ref: MyLaunchTemplate
    Version:
      Fn::GetAtt: "MyLaunchTemplate.LatestVersionNumber"
  MinSize: 1
  MaxSize: 2
  AvailabilityZones:
    Fn::GetAZs: ""

```

Stacks (1)

Filter by stack name

Active View nested

test4

2020-05-13 01:11:32 UTC+0530

CREATE_COMPLETE

Events (8)

Search events

Timestamp	Logical ID	Status	Status reason
2020-05-13 01:12:58 UTC+0530	test4	CREATE_COMPLETE	-
2020-05-13 01:12:57 UTC+0530	MyASG	CREATE_COMPLETE	-
2020-05-13 01:11:39 UTC+0530	MyASG	CREATE_IN_PROGRESS	Resource creation Initiated
2020-05-13 01:11:38 UTC+0530	MyASG	CREATE_IN_PROGRESS	-

13.232.64.89

←

→

↺

🏠

...

🔖

☆

⬇

⌵

🖨

🔍

🔧

🔌

🔄

📦

☰

This is

Changed File

5. Perform ASG Rolling Update with the change in UserData in above Cloudformation Template

```
Resources:
  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Metadata:
      AWS::CloudFormation::Init:
        config:
          packages:
            yum:
              nginx: []
          files:
            "/var/www/html/index.html":
              content:
                Sub: |
                  <br><br>
                  <center>This is <h1>Custom File</h1></center>
              mode: '000755'
              owner: ec2-user
              group: ec2-user
    Properties:
      LaunchTemplateName: "Test-Template"
      LaunchTemplateData:
        InstanceType: t2.micro
        KeyName: Jay
        ImageId: ami-04b2519c83e2a7ea5
        SecurityGroupIds:
          - sg-057ad8e5c2a83dd43
        UserData:
          Fn::Base64:
            Fn::Sub: |
              #!/bin/bash
              yum update -y aws-cfn-bootstrap
              /opt/aws/bin/cfn-init -s ${AWS::StackId} -r MyLaunchTemplate --region ${AWS::Region} || error_exit 'Failed to run cfn-init'
              rm /usr/share/nginx/html/index.html
              cp /var/www/html/index.html /usr/share/nginx/html/
              service nginx start

  MyASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      LaunchTemplate:
        LaunchTemplateId:
          Ref: MyLaunchTemplate
        Version:
          Fn::GetAtt: "MyLaunchTemplate.LatestVersionNumber"
      MinSize: 1
      MaxSize: 2
      AvailabilityZones:
        Fn::GetAZs: ""
      UpdatePolicy:
        AutoScalingRollingUpdate:
          MinInstanceInService: 1
          MaxBatchSize: 2
          WaitOnResourceSignals: 'true'
          PauseTime: PT10M
```


Filter:

<<

<

1 to 1 of 1 Auto Scaling Groups

>

>>

<input checked="" type="checkbox"/>	Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check
<input checked="" type="checkbox"/>	test5-MyASG-...	Test-Template	1	1	1	2	ap-south-1c, ap-south-1b, a...	300	0

Auto Scaling Group: test5-MyASG-EWD6CJ2EV5PO



- Details
- Activity History
- Scaling Policies
- Instances
- Monitoring
- Notifications
- Tags
- Scheduled Actions
- Lifecycle Hooks

Actions



Filter: **Any Health Status**

Any Lifecycle State

<<

<

1 to 1 of 1 Instances

>

>>

<input type="checkbox"/>	Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
<input type="checkbox"/>	i-0827427adbf68a620	InService	Test-Template	ap-south-1b	Healthy	