Cloudformation

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

index.html | vpc-igw-subnet-routetable-route-sg-ec2.yaml

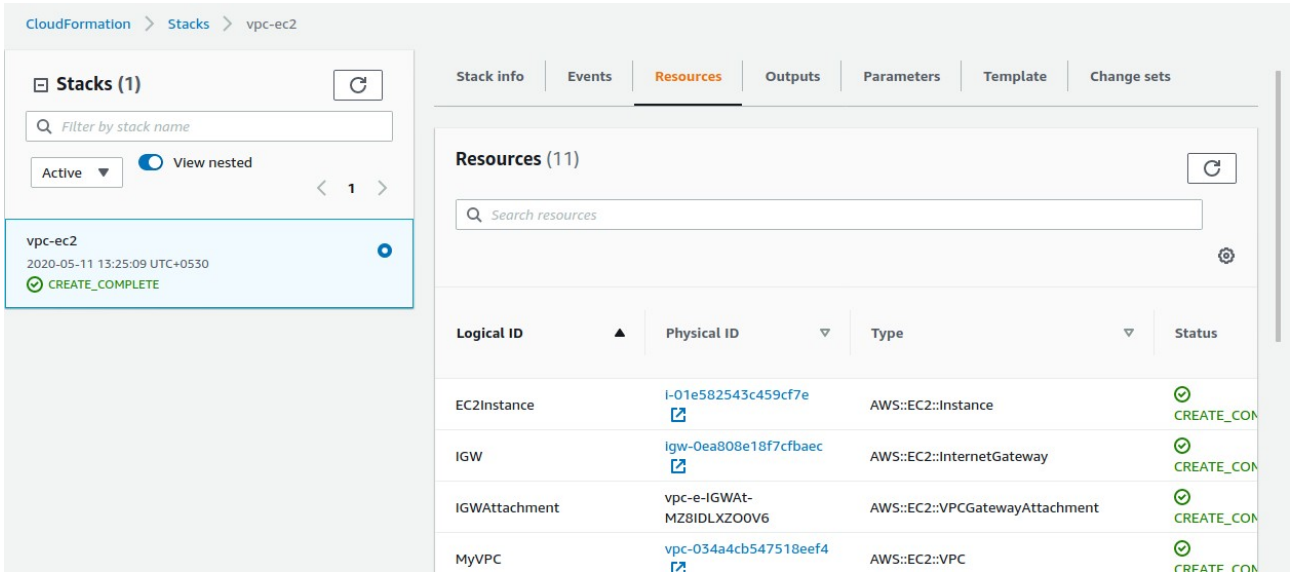
```yaml
Resources:
  MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: owner
          Value: Jay
        - Key: purpose
          Value: Bootcamp
  IGW:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: owner
          Value: jay
        - Key: purpose
          Value: bootcamp
  IGWAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId:
        Ref: IGW
      VpcId:
```

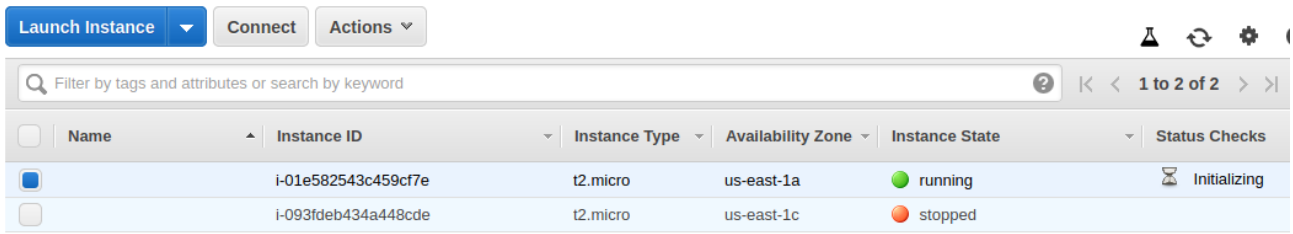
```
      Ref: MyVPC
  PublicSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: MyVPC
      AvailabilityZone: "us-east-1a"
      CidrBlock: 10.0.1.0/24
      MapPublicIpOnLaunch: true
      Tags:
        - Key: owner
          Value: jay
        - Key: purpose
          Value: bootcamp
  PublicSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId:
        Ref: MyVPC
      AvailabilityZone: "us-east-1b"
      CidrBlock: 10.0.2.0/24
      MapPublicIpOnLaunch: true
      Tags:
        - Key: owner
```

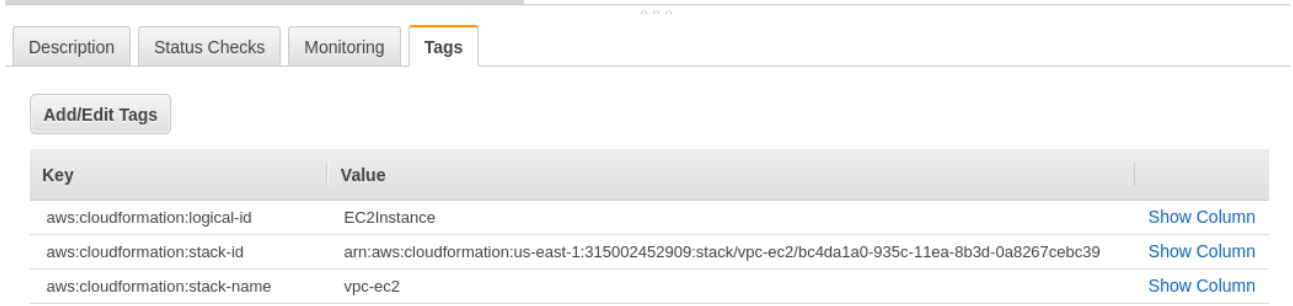
```
        Value: bootcamp
  PublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId:
        Ref: MyVPC
  PublicRoute:
    Type: AWS::EC2::Route
    DependsOn: IGWAttachment          # If IGW is attached with VPC, then onl
    Properties:
      RouteTableId:
        Ref: PublicRouteTable
      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
```







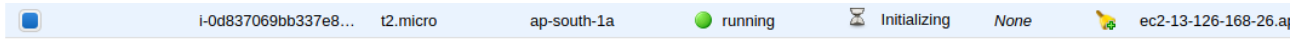
2. Instll Nginx in EC2 and put it in a ASG.
i) First by Userdata
ii) By Metadata

By userdata

```yaml
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
```

```yaml
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" th
        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	-



Instance: i-0d837069bb337e8e2 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:ec2launchtemplate:id	lt-041748c0c283e9dfd	Show Column

View/Change User Data

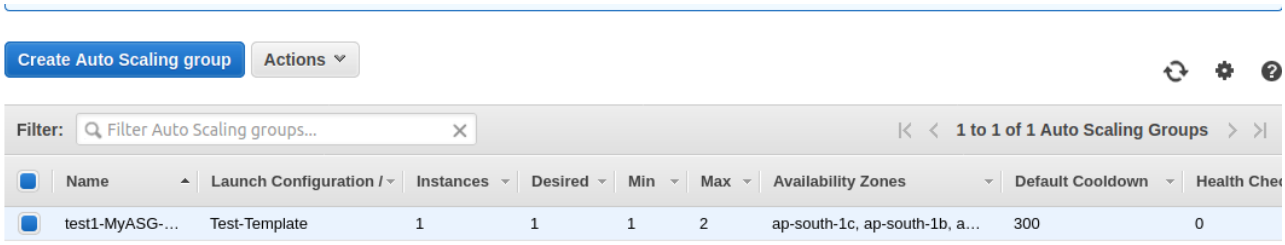
Instance ID: i-0d837069bb337e8e2

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.

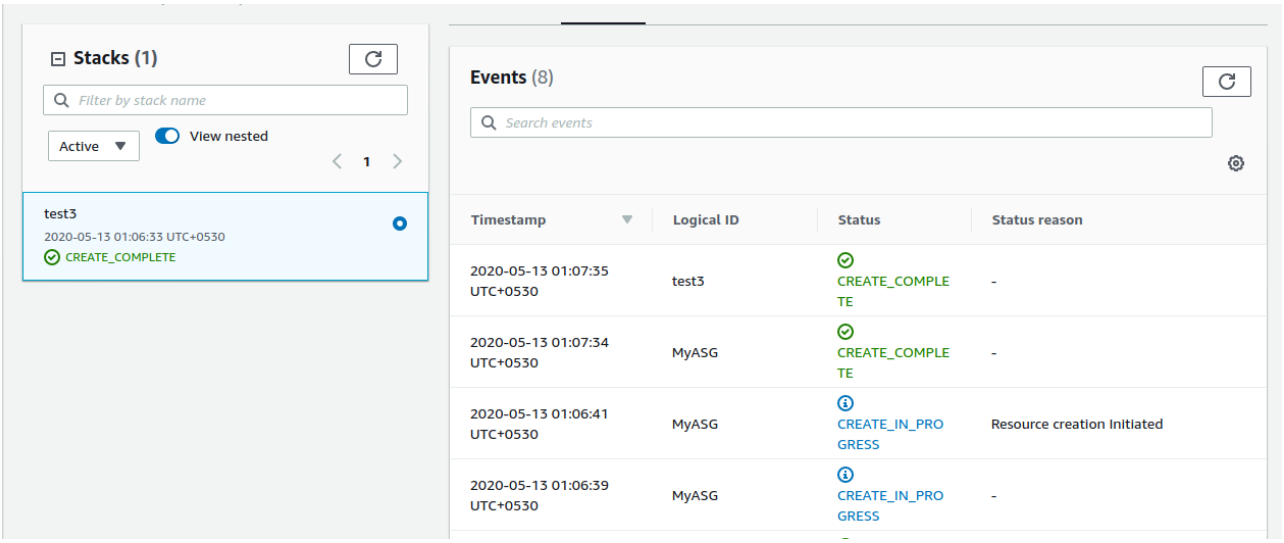
Cancel Save



By metadata

```yaml
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::
              /opt/aws/bin/cfn-hup || error_exit "Failed to start cfn-hup"
              /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackId} --resource MyLaunchTemp
```

```
        - 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 MyLaunchTem
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
```



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: Filter Auto Scaling groups...

1 to 1 of 1 Auto Scaling Groups

Name	Launch Configuration /	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Chec
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: ""
```

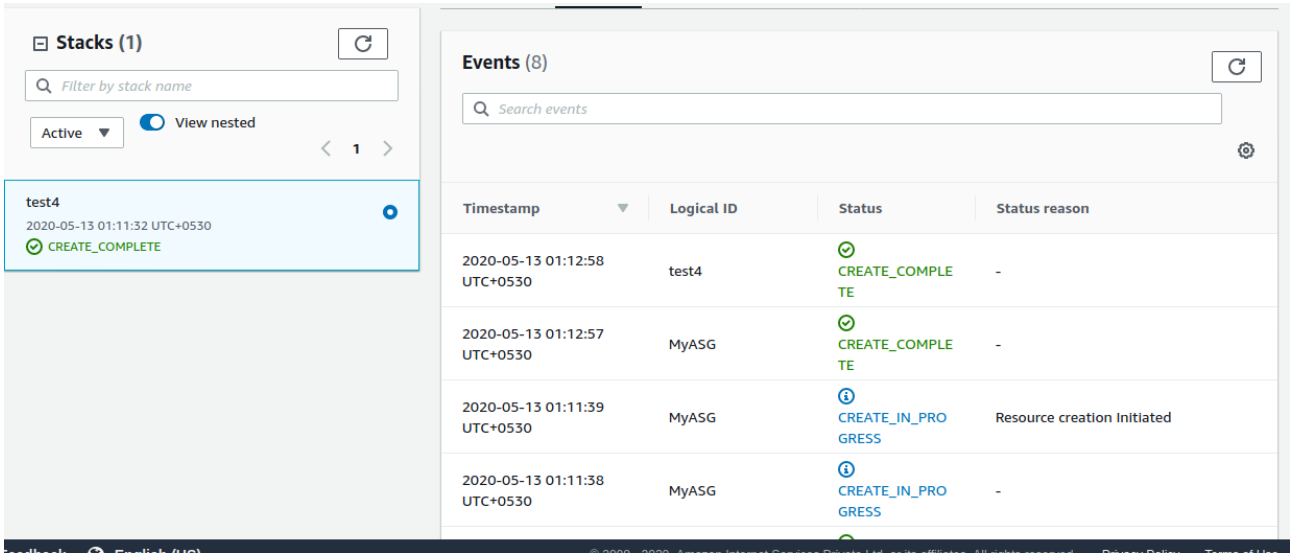
```
[ec2-user@ip-172-31-33-121 ~]$ cd /var/www/html/
[ec2-user@ip-172-31-33-121 html]$ ll
total 4
-r-------- 1 root root 68 May 12 19:20 index.html
[ec2-user@ip-172-31-33-121 html]$ cat index.html
cat: index.html: Permission denied
[ec2-user@ip-172-31-33-121 html]$ sudo cat index.html
{"Sub": "<br><br>\n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

```yaml
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":
```

```yaml
              content:
                Fn::Sub: |
                  [cfn-auto-reloader-hook]
                  triggers=post.upadte
                  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: ""
```



This is

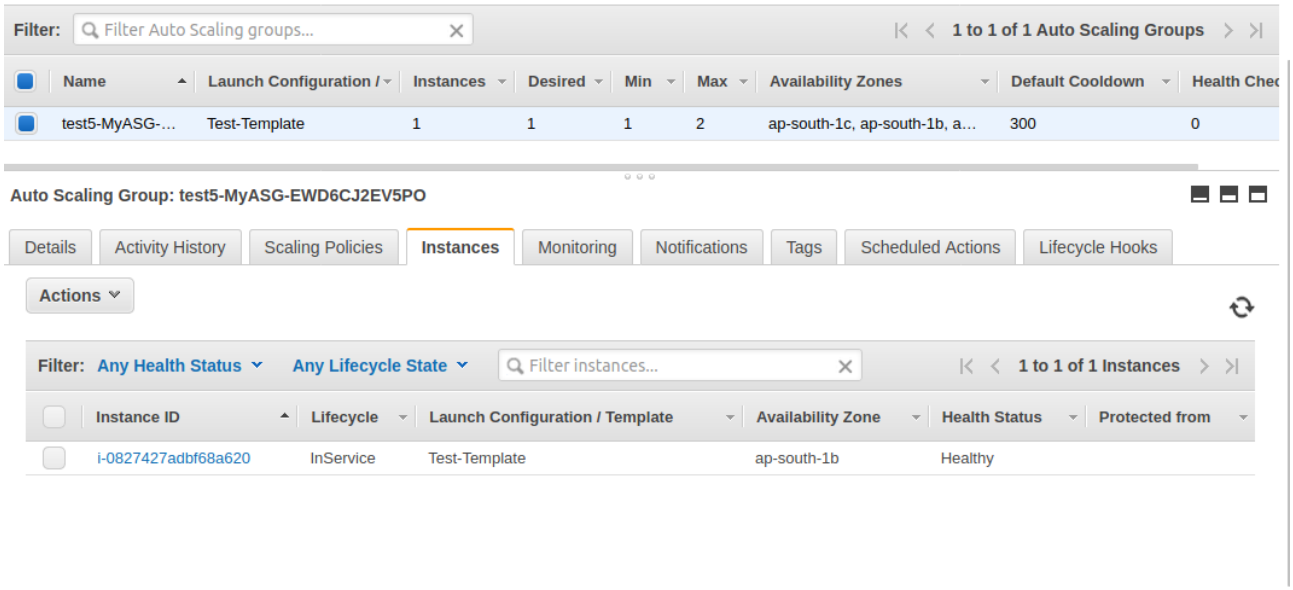
Changed File

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

```yaml
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-i
              rm /usr/share/nginx/html/index.html
              cp /var/www/html/index.html /usr/share/nginx/html/
              service nginx start
  MyASG:
```

```yaml
  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
```

Column 33


Filter:
Filter Auto Scaling groups...
1 to 1 of 1 Auto Scaling Groups
Name
Launch Configuration /
Instances
Desired
Min
Max
Availability Zones
Default Cooldown
Health Chec
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
Filter instances...
1 to 1 of 1 Instances
Instance ID
Lifecycle
Launch Configuration / Template
Availability Zone
Health Status
Protected from
i-0827427adbf68a620
InService
Test-Template
ap-south-1b
Healthy