😊 Resource EC2 Instance

In this lab, the goal is to create and manage an EC2 instance using AWS CloudFormation. The process includes the following key steps:

1. **Creating an EC2 Instance**: Using a CloudFormation template, you create an EC2 instance that includes a default security group. You'll need an AMI ID and a Key Pair for this setup.
2. **Using CloudFormation Templates**: You download, unzip, and modify templates from GitHub, then upload them to create a stack in CloudFormation.
3. **Managing Security Groups**: You create a new security group and attach it to the instance using CloudFormation.
4. **Updating Security Groups**: You add new rules to the existing security group by modifying and uploading the template.
5. **Adding Elastic IP**: You associate an Elastic IP with the instance by updating the CloudFormation template.

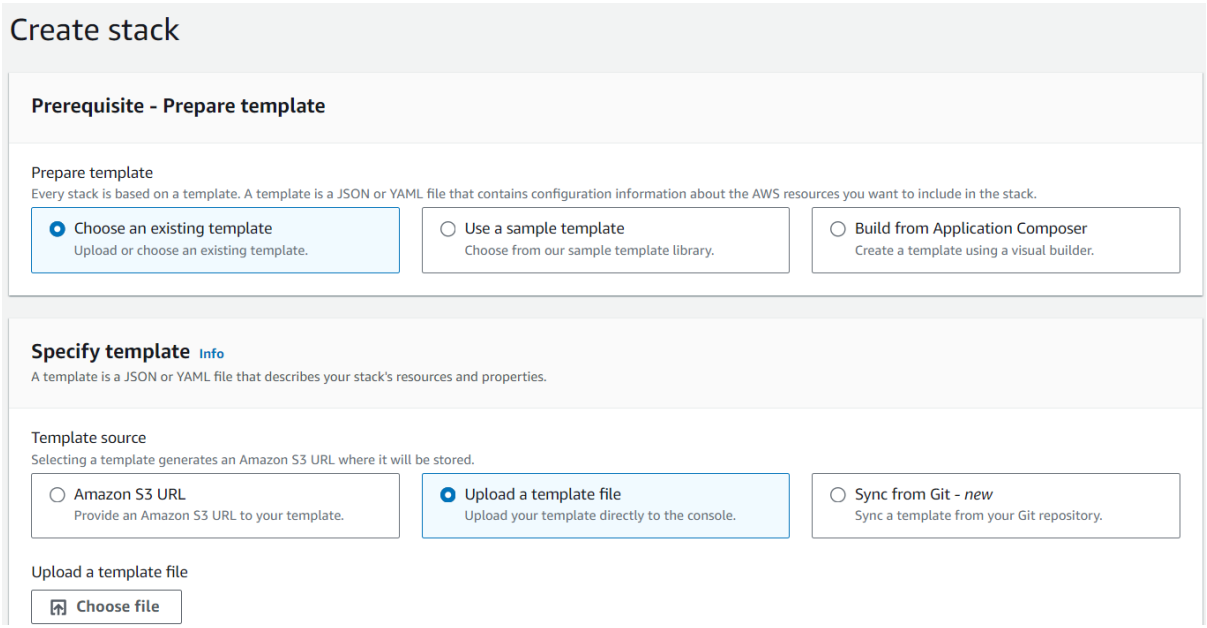
End Goal: The primary objective is to automate the creation, configuration, and management of an EC2 instance using CloudFormation, demonstrating how to manage infrastructure as code, modify security settings, and allocate resources efficiently.

😄 To begin with the Lab

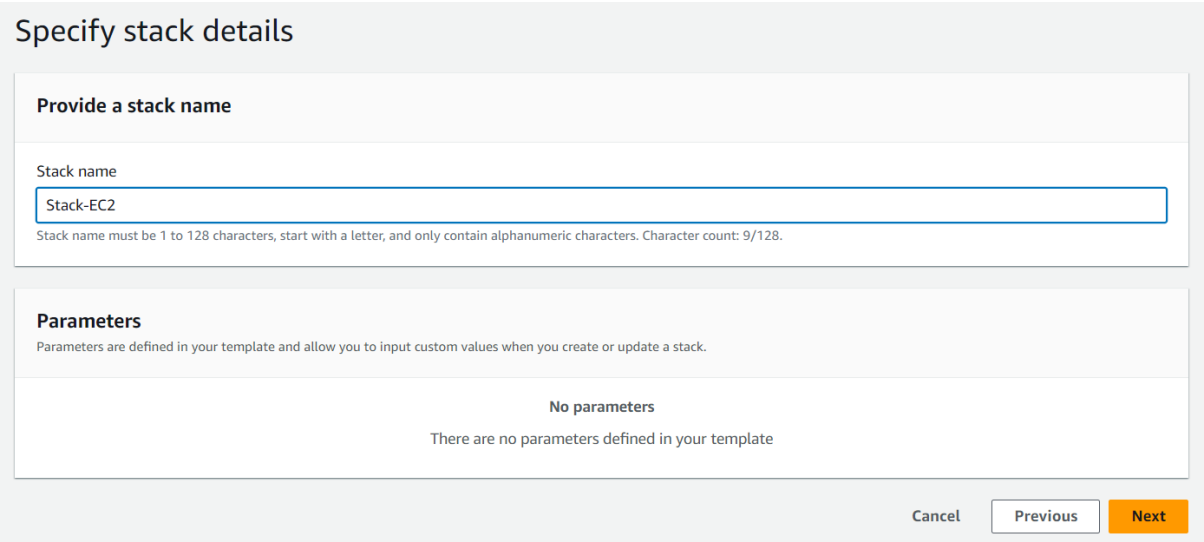
1. In this lab we are going to create an EC2 instance using cloud formation. Below you can see the code that we are going to use. Here you can see that we are including the security group, which is the default SG for now.
2. Now you can get the templates from GitHub. Download them and unzip them make changes accordingly and then use them.
3. Also, you need to have the AMI ID and the Key pair. For key pair go to EC2 and create it and for AMI ID, you can get it by browsing.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Basic EC2 Instance.

Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0e97ea97a2f374e3d
      InstanceType: t2.micro
      KeyName: CFKeyPair
      SecurityGroups:
        - default
```

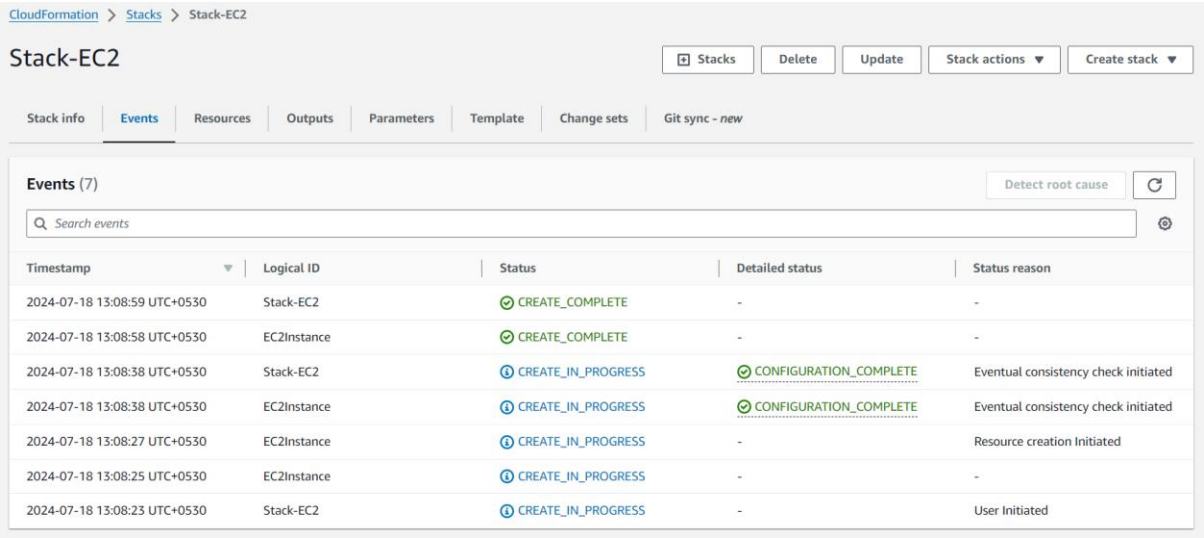
4. Now in your AWS Console you need to navigate to cloud formation and then you need to create a stack.
5. Then you need to click on choose an existing template then we need to choose to upload a template file. After that browse into your system and upload the template.



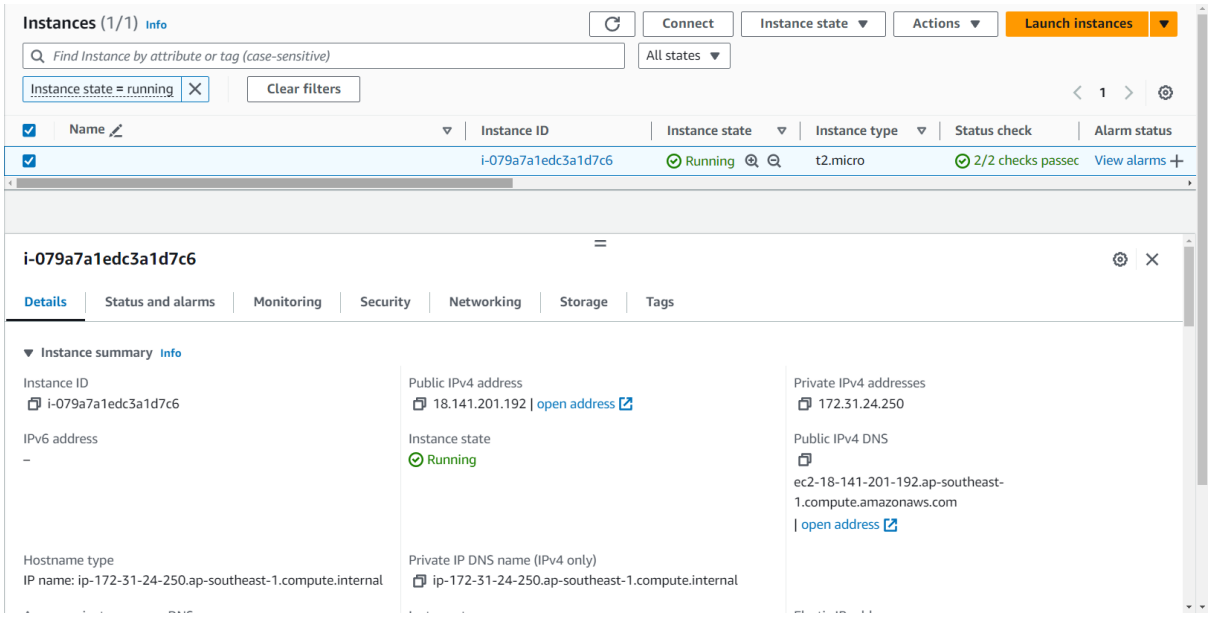
6. Now you need to give your stack a name then moves to the review page and create your stack.



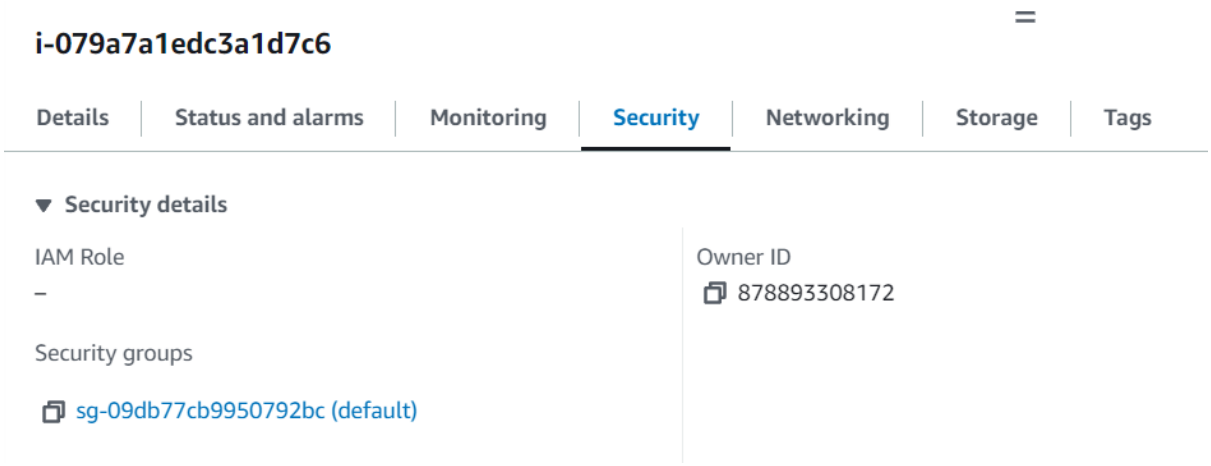
7. Now if you go to the Events section then you will see the events which can help you to visualize the process scenarios.



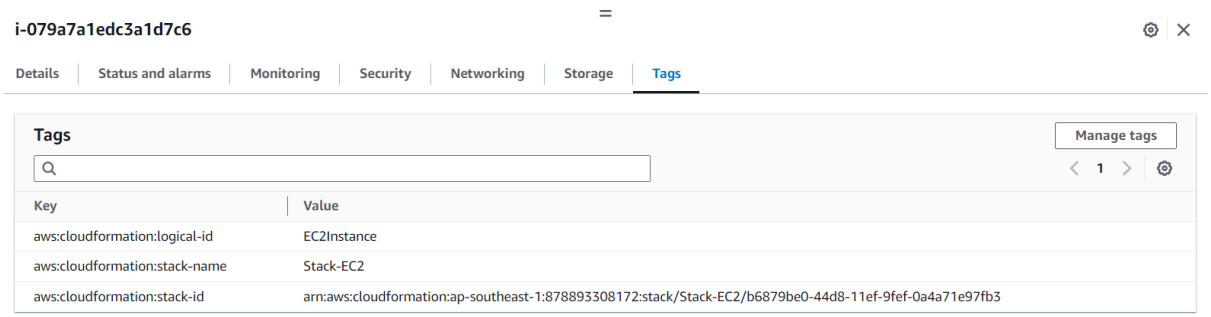
8. Also, navigate to EC2 and here you will see that our instance has been created. You can check for the key pair.



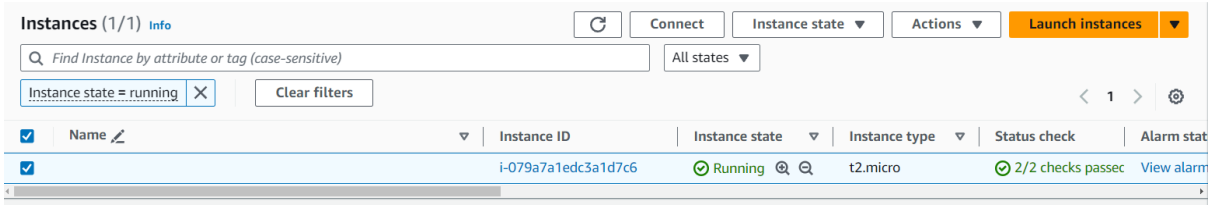
9. Now in the security you will see your security group attached to your instance.

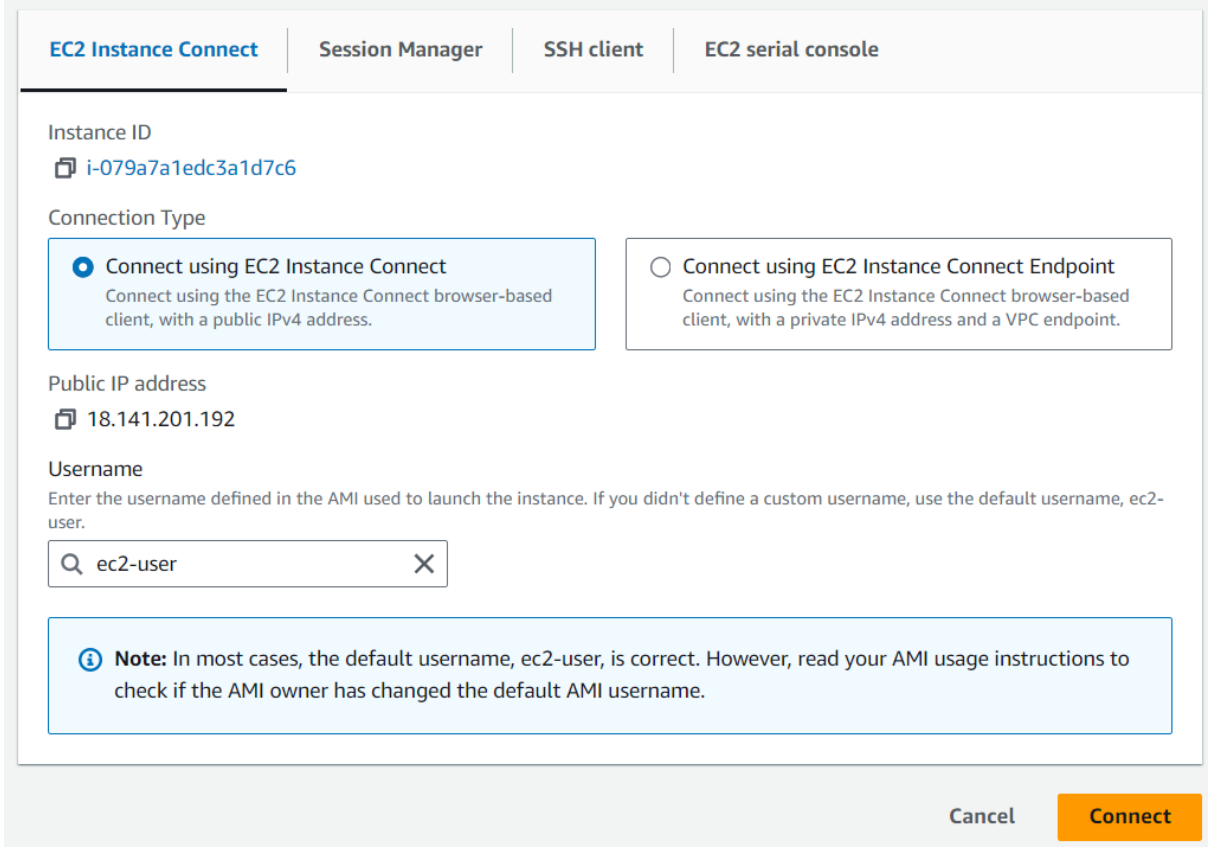


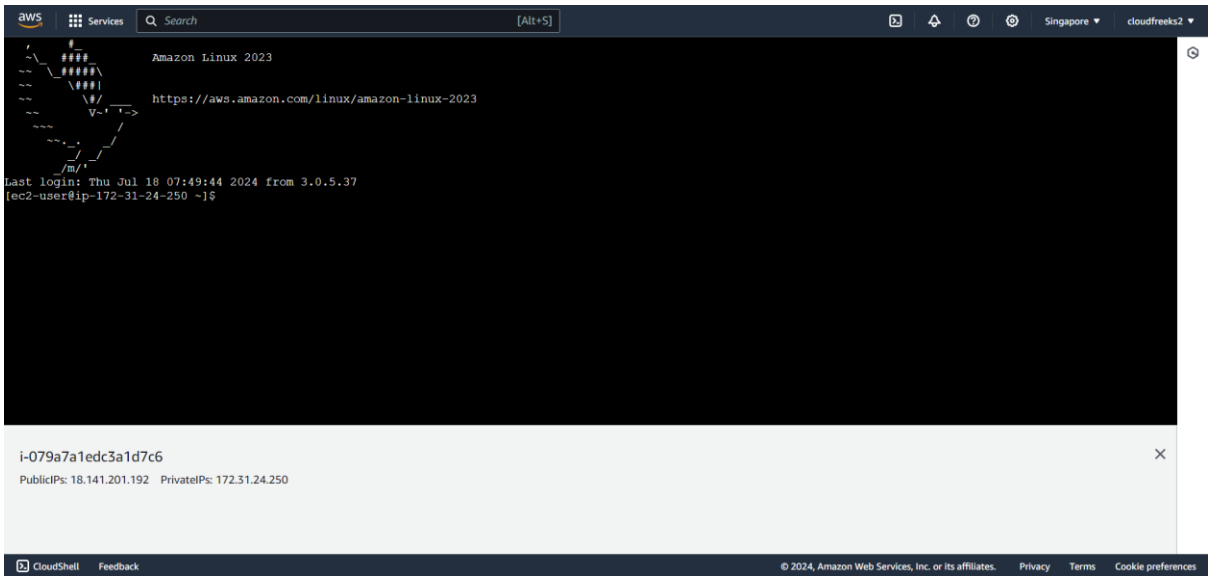
10. Also, you can view the different tags assigned to your instance.



11. Now if you click on connect and choose EC2 instance connect and try to connect your instance you will see that your instance was successfully connected.





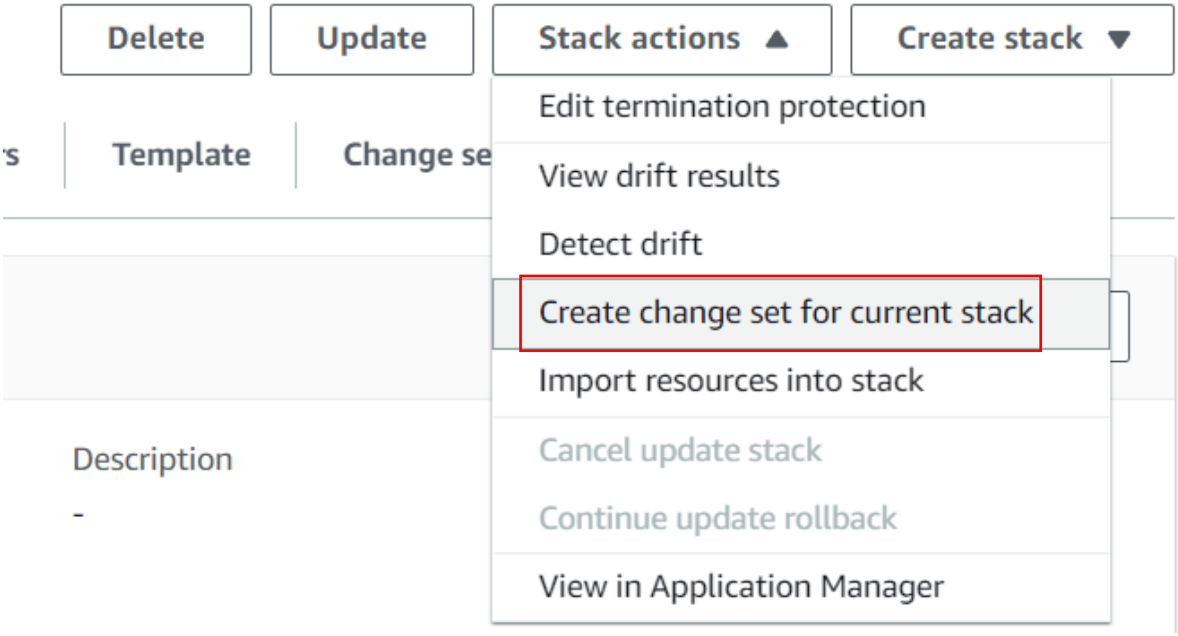


😄 Security Group

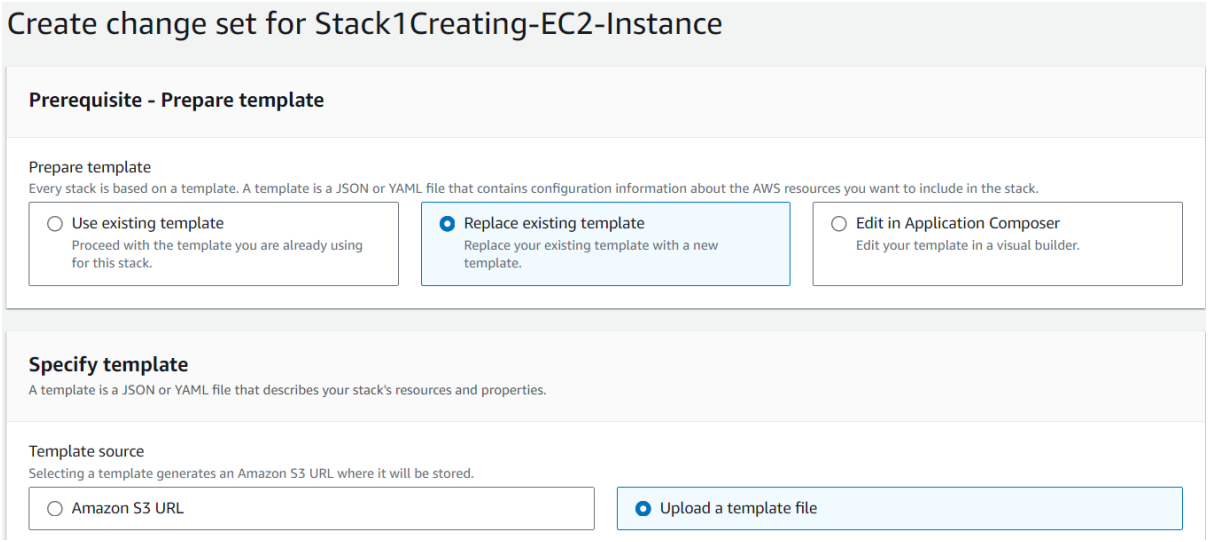
1. Now we will create a new security group for our instance and attach it. Below you can see that code that we are using in our template.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Basic EC2 Instance.

Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0e97ea97a2f374e3d
      InstanceType: t2.micro
      KeyName: CFKeyPair
      SecurityGroups:
        - default
        - !Ref SSHSecurityGroup
  SSHSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: my new SSH SG
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '22'
          ToPort: '22'
          CidrIp: 0.0.0.0/0

```

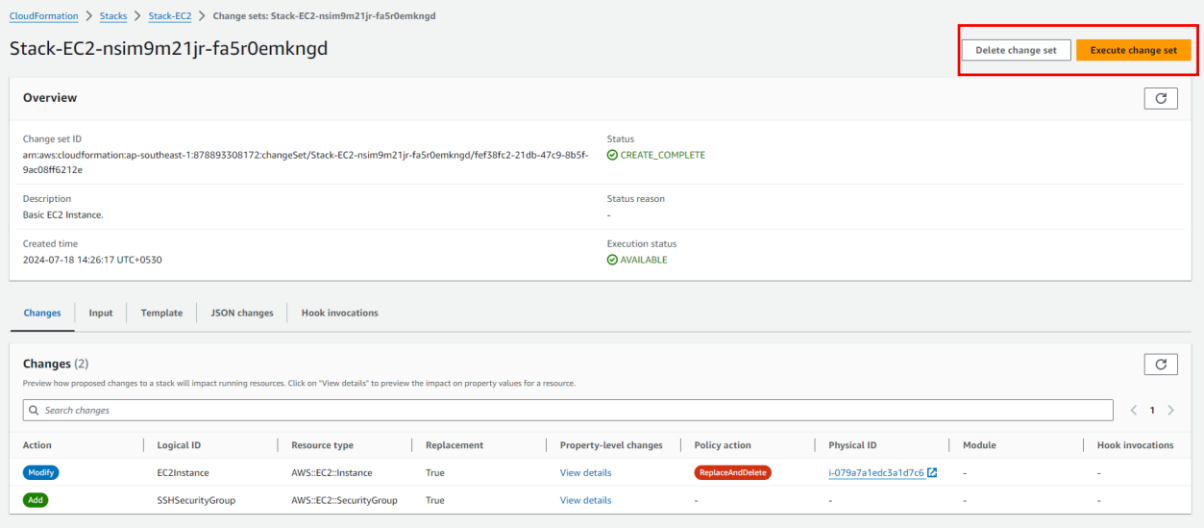
2. So, now in your cloud formation you need to expand stack actions and choose to create a change set for the current stack.



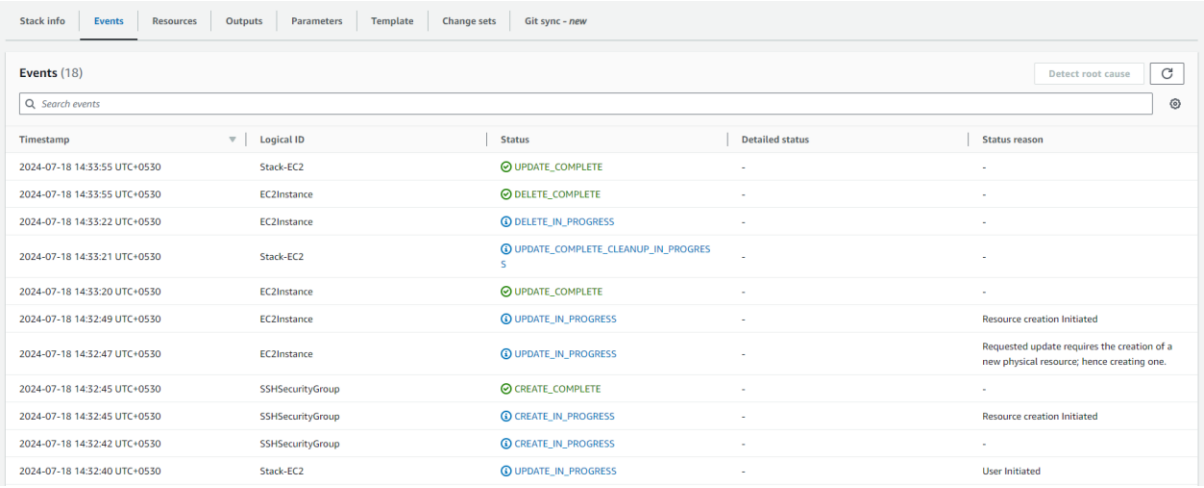
3. And again, we need to replace it with the existing template and then upload our template.
4. After that just move to the review page and click on submit.



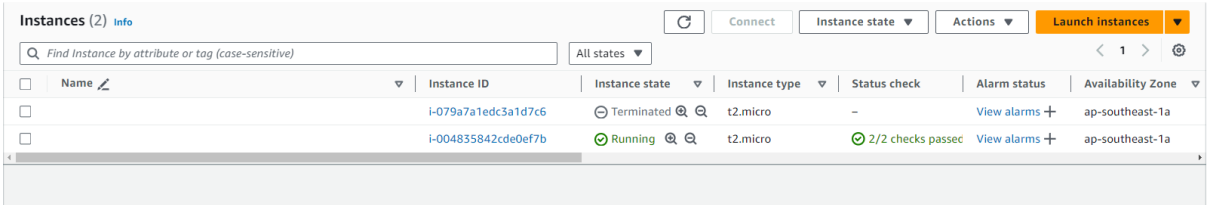
5. Now you will see that your change set has been created and now you have the ability to execute this change set.
6. If you want to execute then click on execute or else, you can delete this change set.
7. Now just click on execute change set. Also, you can see that there are a total of two changes.



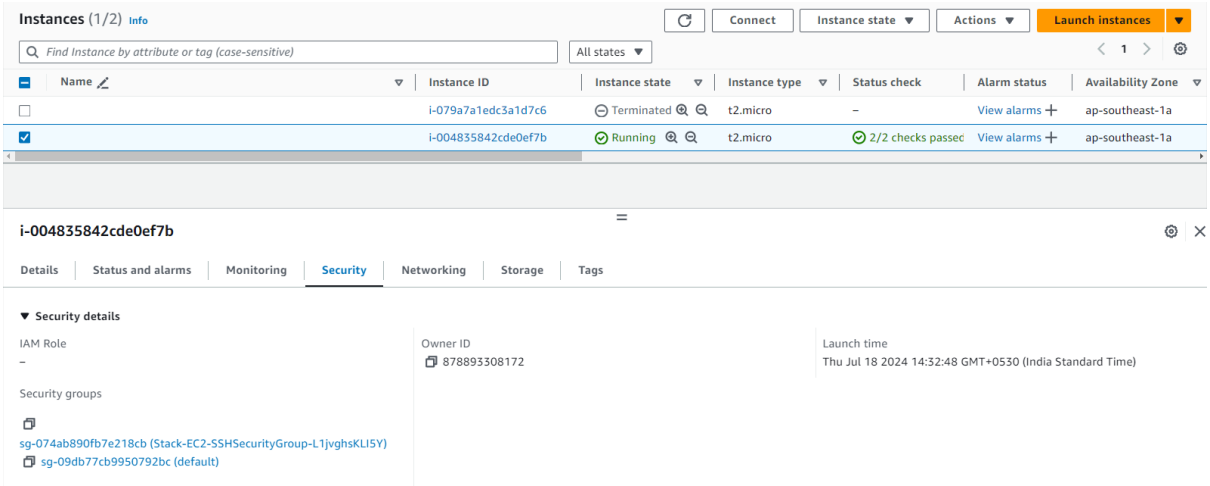
8. Then go to the events to see the real-time progress and below you can see the process.



9. Now go to EC2, here you will see that our previous instance was terminated and the new instance was created.



10. In your new instance if you go to the security tab you will see two security groups also you can log in to your instance through the Putty application.

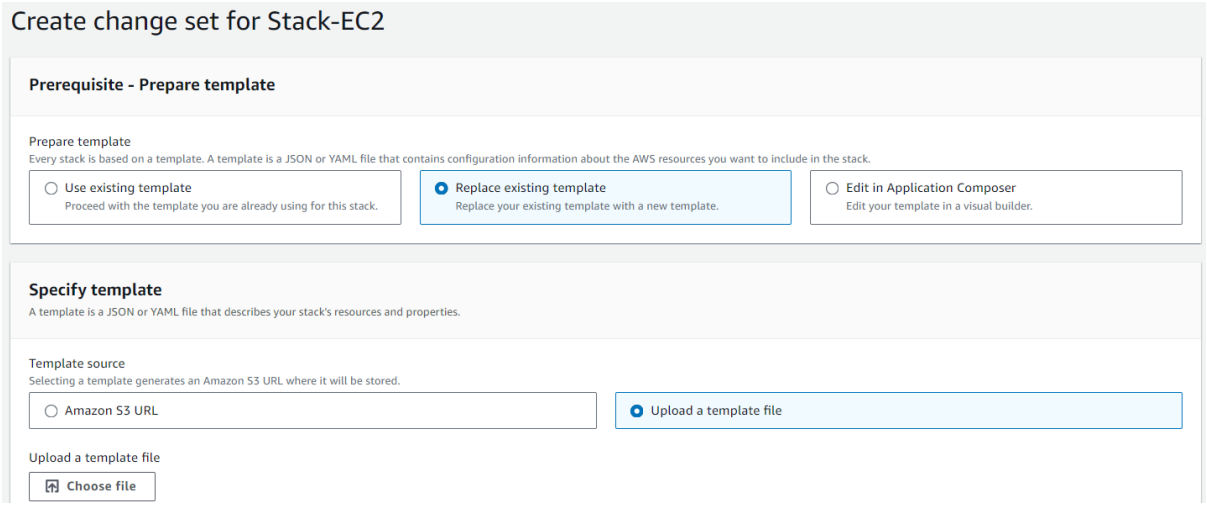


😄 Adding a new rule

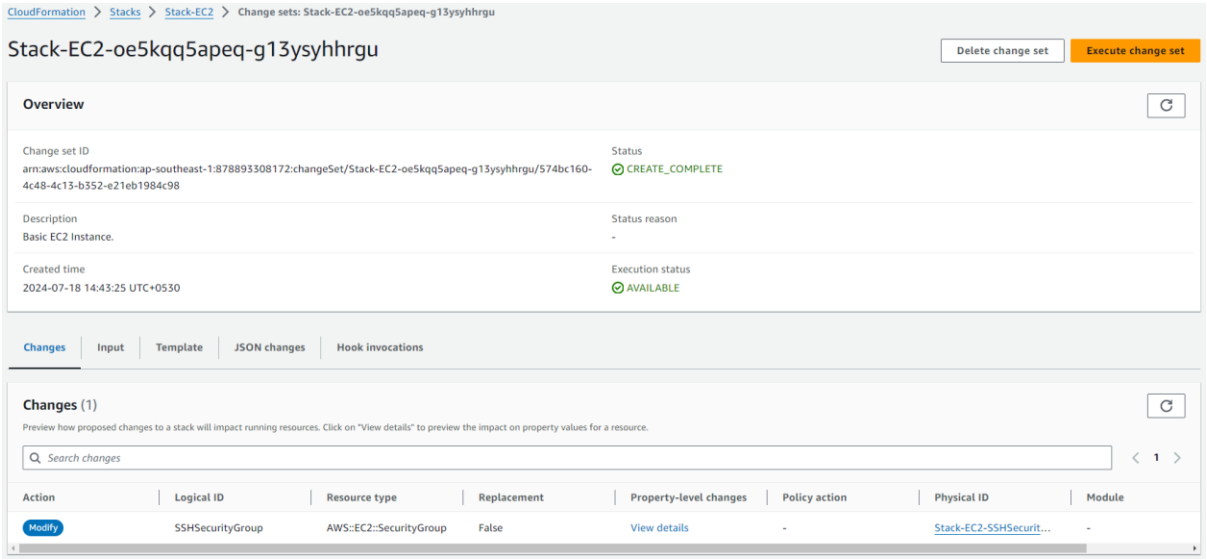
1. Now we are going to add a new rule to our existing security group. For that, we have added some changes to our template. Below you can see that from line 23 we have introduced a new port.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Basic EC2 Instance.

Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0e97ea97a2f374e3d
      InstanceType: t2.micro
      KeyName: CFKeyPair
      SecurityGroups:
        - default
        - !Ref SSHSecurityGroup
  SSHSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: my new SSH SG
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '22'
          ToPort: '22'
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: '8080'
          ToPort: '8080'
          CidrIp: 0.0.0.0/0
```

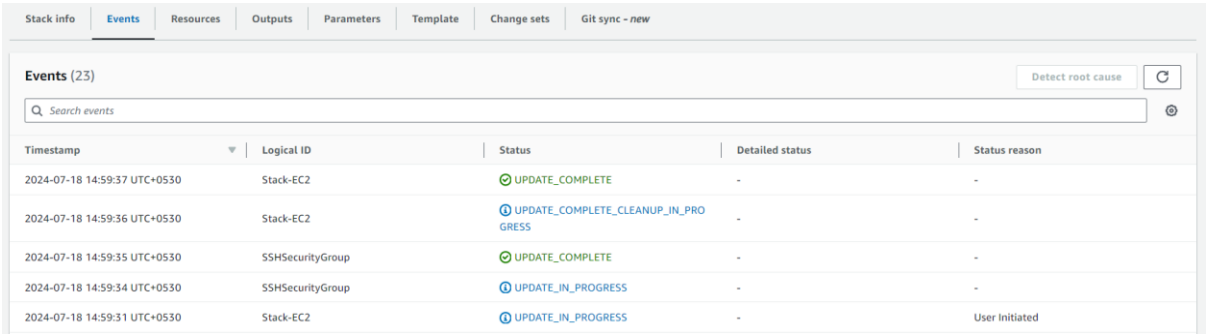
2. Now again we need to go to our Cloud formation stack and choose the create change stack feature to upload our new template.



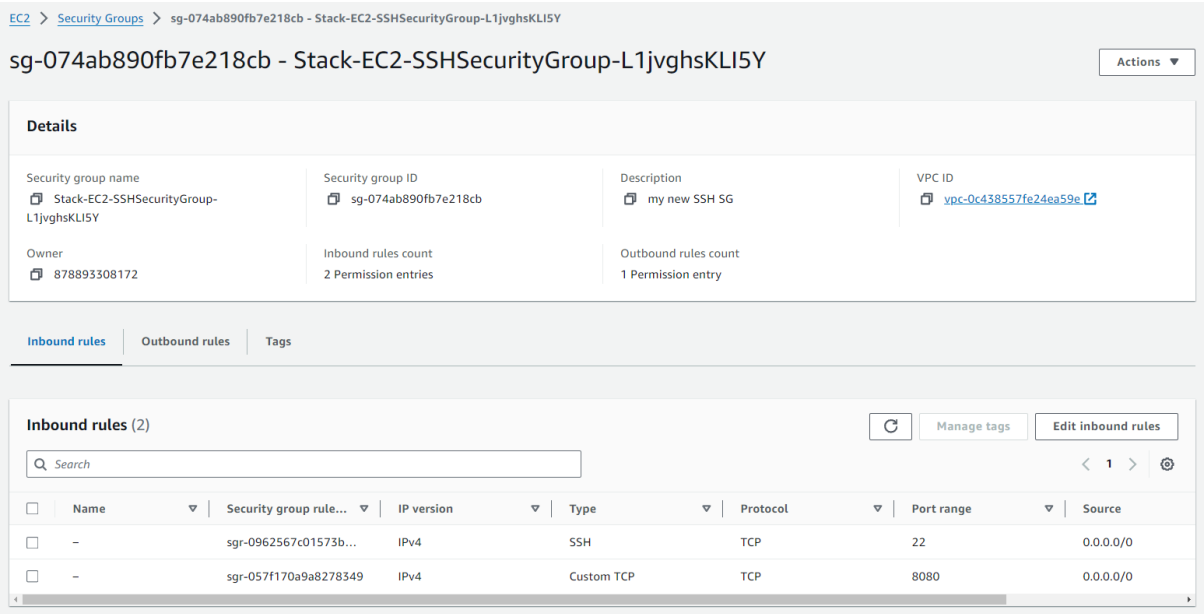
3. This time you can see that there is only one change that will occur.



4. Now in the events tab you can see that our update has bene initiated now if we go to our security group we will see the added inbound rule.



5. Below we can see that the port has been added successfully.

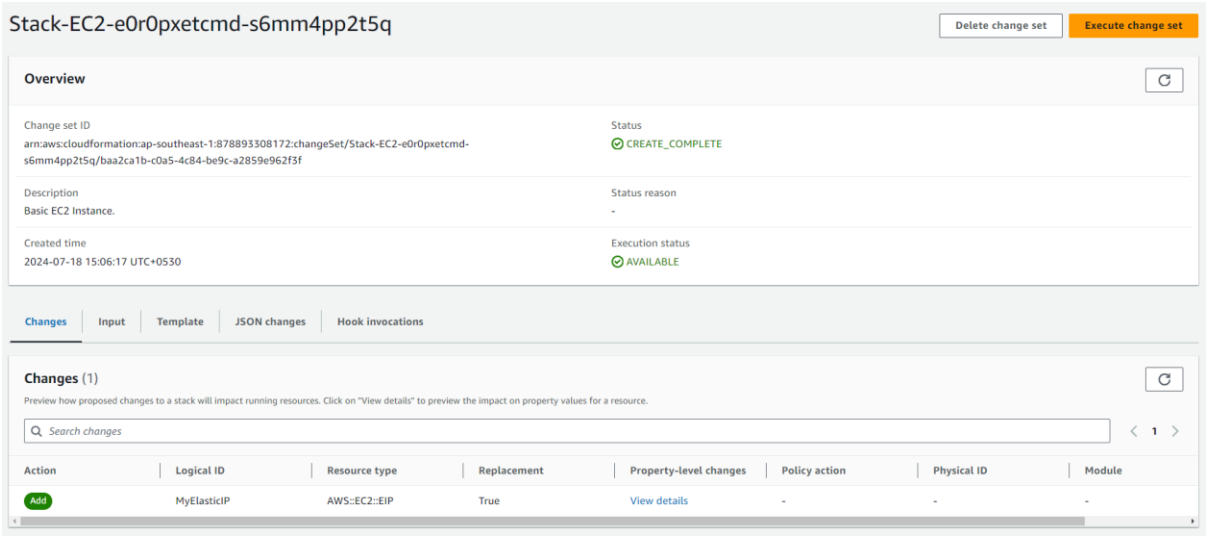


😄 Adding Elastic IP

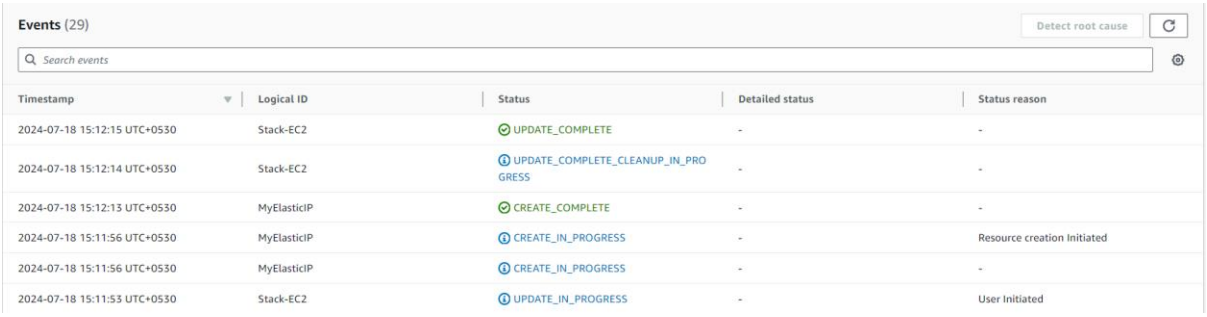
1. Now we are going to add an Elastic IP to our instance, for that we have our template ready. You can see the code below.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Basic EC2 Instance.

Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-0e97ea97a2f374e3d
      InstanceType: t2.micro
      KeyName: CFKeyPair
      SecurityGroups:
        - default
        - !Ref SSHSecurityGroup
  SSHSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: my new SSH SG
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: '22'
          ToPort: '22'
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: '8080'
          ToPort: '8080'
          CidrIp: 0.0.0.0/0

  MyElasticIP:
    Type: AWS::EC2::EIP
    Properties:
      InstanceId: !Ref EC2Instance
```

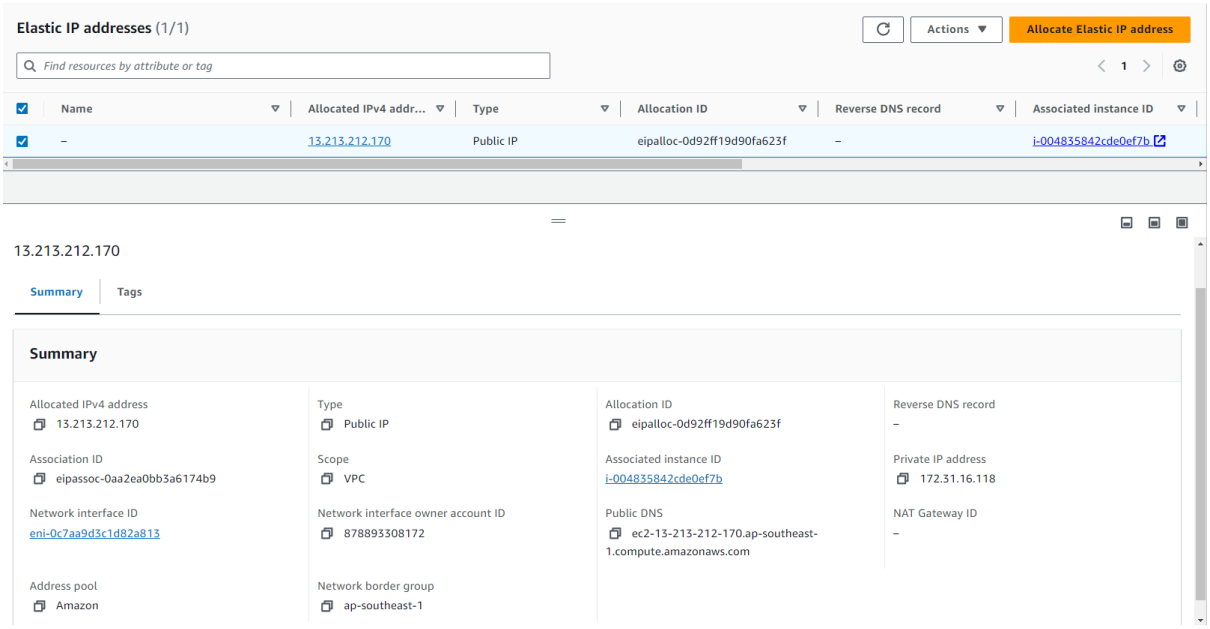
2. Now again the steps are the same just go to your stack and choose to create a change stack set for the current stack.
3. Below you can see that it is showing us the addition of our Elastic IP.



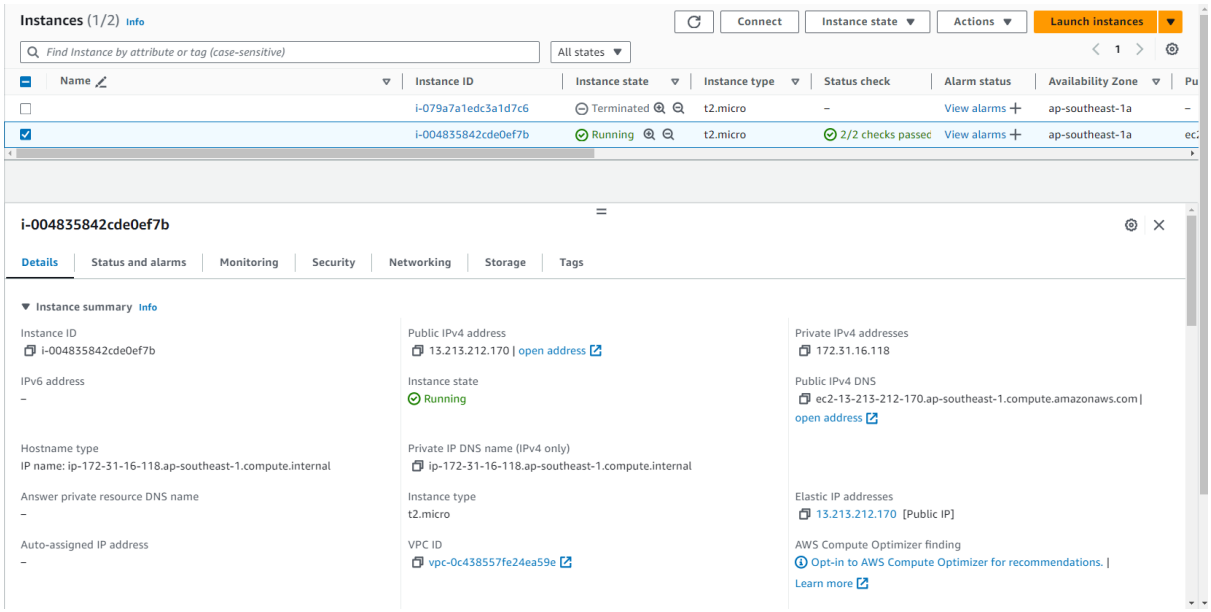
4. Also, in the events you can see the process.



5. Now go to EC2 and check whether your instance has the Elastic IP or not. First, go directly to Elastic IP and here you can see that, then go to the associated instance ID.



6. And in the instance too you will see the elastic IP address.


Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
i-079a7a1edc3a1d7c6
Terminated
t2.micro
–
View alarms
ap-southeast-1a
i-004835842cde0ef7b
Running
t2.micro
2/2 checks passed
View alarms
ap-southeast-1a
i-004835842cde0ef7b
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-004835842cde0ef7b
Public IPv4 address
13.213.212.170 | open address
Private IPv4 addresses
172.31.16.118
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-13-213-212-170.ap-southeast-1.compute.amazonaws.com | open address
Hostname type
IP name: ip-172-31-16-118.ap-southeast-1.compute.internal
Private IP DNS name (IPv4 only)
ip-172-31-16-118.ap-southeast-1.compute.internal
Answer private resource DNS name
–
Instance type
t2.micro
Elastic IP addresses
13.213.212.170 [Public IP]
Auto-assigned IP address
–
VPC ID
vpc-0c438557fe24ea59e
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more