**COMP4964 – DevOps Engineering**

**Lab 9**

**CloudFormation**

In this lab, you will learn CloudFormation as a tooling in AWS.

Part 1: CloudFormation Exploration

1. Log in your AWS account.
2. Go to CloudFormation console. You may search for it or use this link: <https://us-west-2.console.aws.amazon.com/cloudformation/>
3. You should see a few existing stacks (from your previous labs)
4. Choose one of your existing stacks

(I chose my “EC2ContainerService-ecs-20480-tile-game” stack, feel free to pick another)

1. Open “Resources” tab

Graphical user interface, application, Word

Description automatically generated

1. Make a screenshot of your Resources tab and paste it below:

|  |
| --- |
|  |

1. Open “Template” tab
2. Make a screenshot of your Template tab and paste it below:

|  |
| --- |
|  |

1. Open designer’s view:

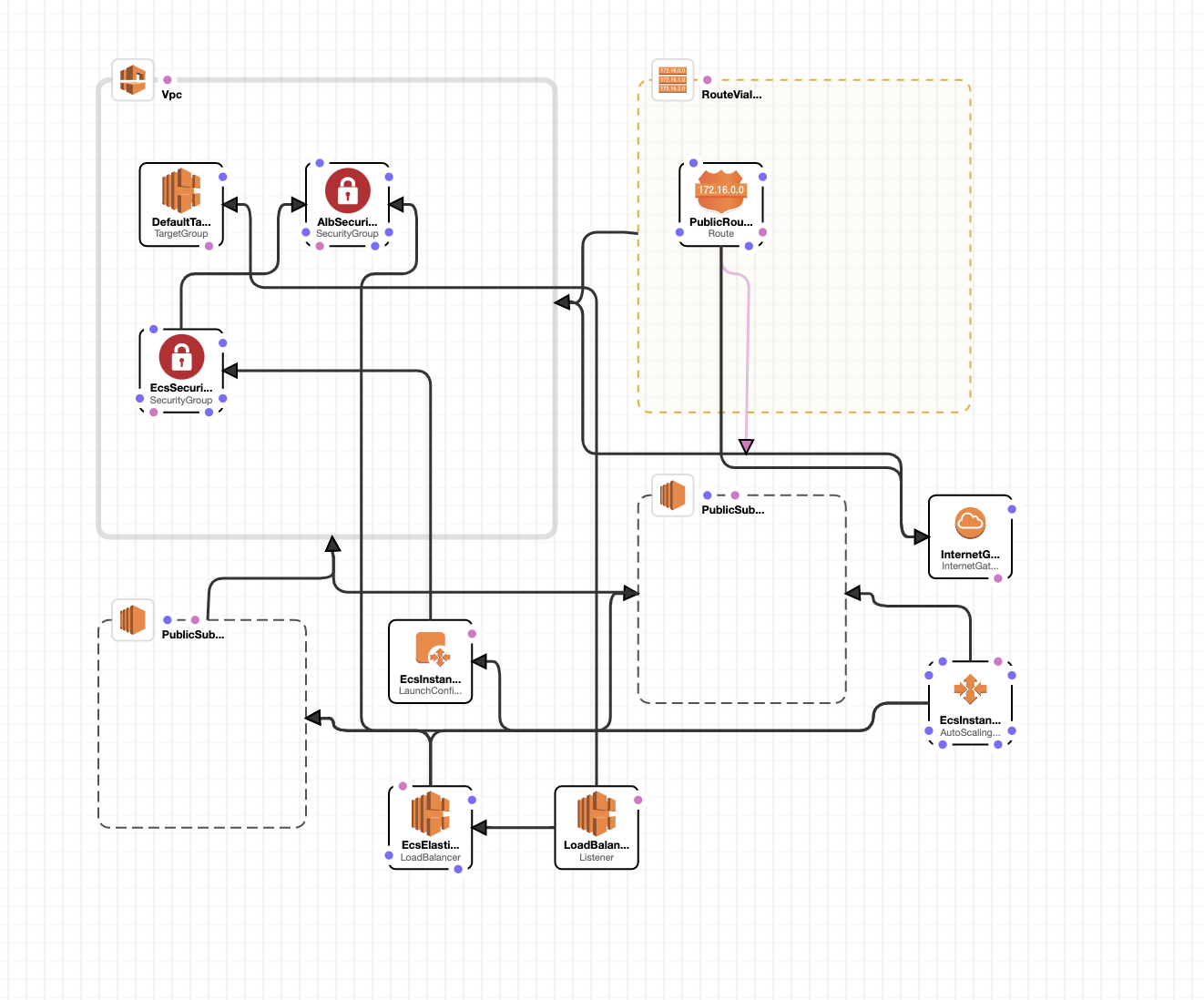
Graphical user interface, application

Description automatically generated

1. Make a screenshot on your stack structure, and paste it below

|  |
| --- |
|  |

Mine is like this. Yours may be different.



1. Play with your view, check out more details for your template.
2. To get sign-off for this part, you need to:
   1. Show me your screenshots
   2. Explain to your instructor more details of your template.

Part 2: Create your stack using CloudFormation

1. Change your region to us-east-1
2. Create an YAML file with the content below

|  |
| --- |
| ---  Resources:  MyInstance:  Type: AWS::EC2::Instance  Properties:  AvailabilityZone: us-east-1a  ImageId: ami-a4c7edb2  InstanceType: t2.micro |

1. Go back to CloudFormation Console >> Stacks
2. Click “Create stack” button

Graphical user interface, text, application, email

Description automatically generated

1. Then choose “with new resources (standard)”
2. For the “Prepare template” section, Choose “Template is ready” (default)
3. Choose “Upload a template file”, and then choose the file you just made.
4. Click “Next”
5. Type in a stack name. One word (Eg, stack-lab)
6. Click “Next”
7. In Step 3, keep everything as defaults.
8. Click “Next”
9. Step 4 is to review your configurations
10. Make a screenshot on your Template URL and paste it below.

|  |
| --- |
|  |

1. Click “Submit”
2. Now you should see detail page of your stack, with showing “Events” tab.

This is what I have:

Graphical user interface, text, application, email

Description automatically generated

1. Refresh the page and monitor the progress, till seeing “CREATE COMPLETE”

Graphical user interface, application

Description automatically generated

1. Go to your EC2 dashboard, you should see a new EC2 instance is running.
2. Show these to your instructor to get your sign-off:
   1. your EC2 instance is running
   2. In Tags tab, show “cloudformation” term, like this:

Graphical user interface, application

Description automatically generated

1. Do not delete your stack nor terminate your EC2 instance, you will need them for the next part.

Part 3: Update your stack using CloudFormation

1. Go to Learning Hub, download the YAML file named 1-ec2-with-sg-eip
2. Go back to your CloudFormation >> Stacks
3. Select your stack, and then click “Update” button

Graphical user interface, text, application, email

Description automatically generated

1. In the “Prepare template” section, choose “Replace current template”.
2. Choose “Upload a template file”
3. Choose the file you just downloaded from Learning Hub. ()
4. “Next”
5. Type in some message for new security group, such as “a new security group from cf”

Graphical user interface, text, application

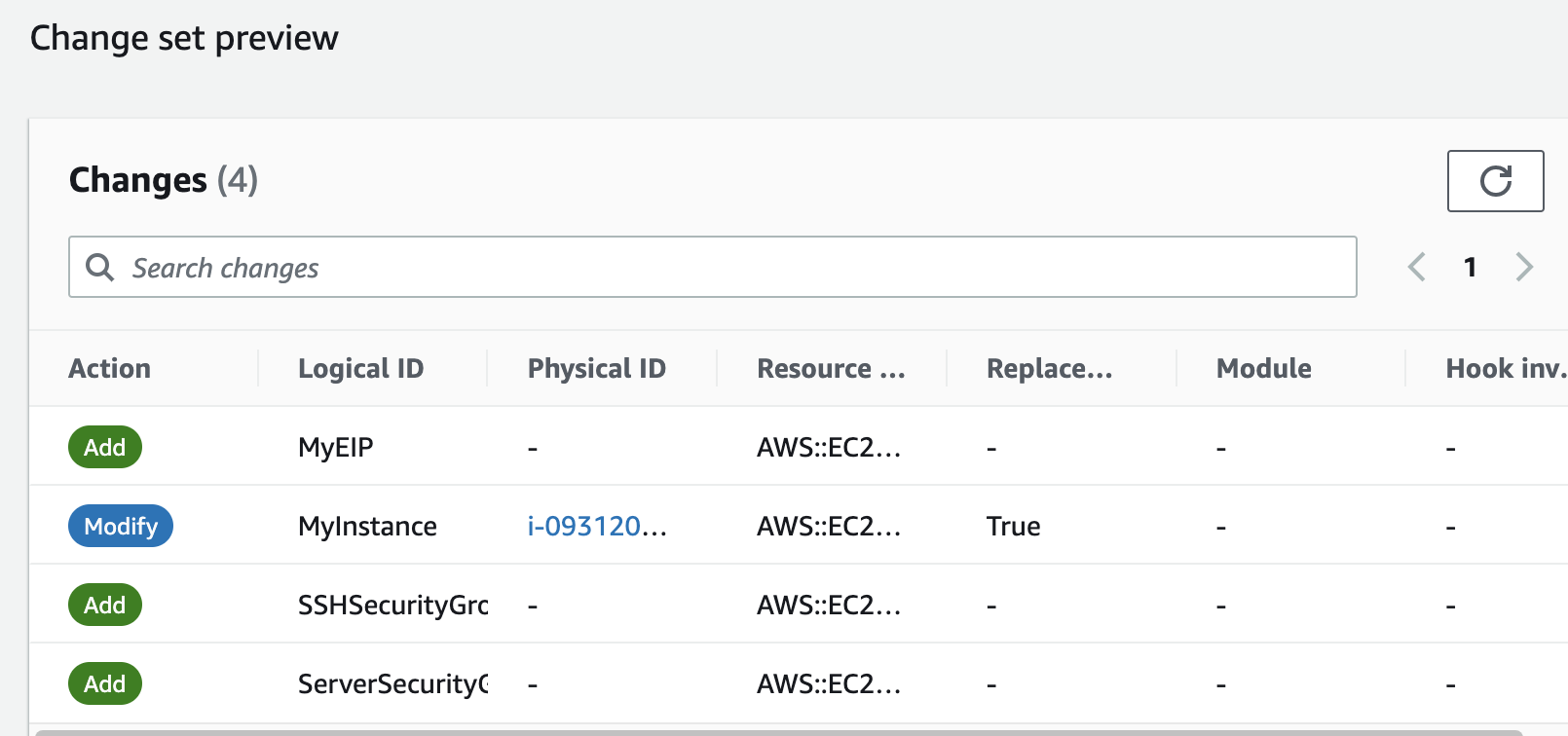
Description automatically generated

1. Click “Next” to Step 3
2. Click “Next” to Step 4
3. At the bottom of Step 4 (Review Page) you should see a table format Change set preview

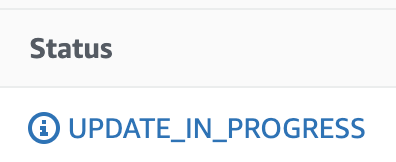
Make a screenshot of it, and paste it below

|  |
| --- |
|  |

Mine looks like below. Yours should be similar



1. Click “Submit”
2. You should see this:



1. Wait till seeing “UPDATE\_COMPLETE”
2. Double check your EC2 instance, EIP, security groups are all in place.
3. Stop here to get a sign off by your instructor, by showing all of:

* EC2 instance list
* EIP
* Security groups with their rules

1. Teardown, after signing off, to delete the stack
2. It should also remove/delete all your resources, make sure to double check.

Part 4: Your turn to make a stack template

Check out the doc on CloudFormation Template:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/gettingstarted.templatebasics.html>

Write 3 template files (YAML files) use validator to check your syntax, and run them to confirm they work. Paste your files contents below

It’s your choice on what your yaml files do. Just make sure they are not too similar.

Hint:

No need to be too complex. You may start thinking of S3 setup or communication, CI/CD, cloudwatch dashboard, lambda etc.

|  |
| --- |
| ---  AWSTemplateFormatVersion: '2010-09-09'  Description: 'CloudFormation template for API Gateway REST API'  Resources:  MyRestApi:  Type: AWS::ApiGateway::RestApi  Properties:  Name: MyRestAPI  Description: Sample REST API  FailOnWarnings: true  EndpointConfiguration:  Types:  - REGIONAL  MyResource:  Type: AWS::ApiGateway::Resource  Properties:  RestApiId: !Ref MyRestApi  ParentId: !GetAtt MyRestApi.RootResourceId  PathPart: 'myresource'  MyMethod:  Type: AWS::ApiGateway::Method  Properties:  RestApiId: !Ref MyRestApi  ResourceId: !Ref MyResource  HttpMethod: GET  AuthorizationType: NONE  Integration:  Type: MOCK  IntegrationResponses:  - StatusCode: 200  RequestTemplates:  application/json: '{"statusCode": 200}' |
| ---  AWSTemplateFormatVersion: '2010-09-09'  Description: 'CloudFormation template for S3 Bucket'  Resources:    MyS3Bucket:      Type: AWS::S3::Bucket      Properties:        BucketName: comp4964-mho-lab9        VersioningConfiguration:          Status: Enabled        PublicAccessBlockConfiguration:          BlockPublicAcls: true          BlockPublicPolicy: true          IgnorePublicAcls: true          RestrictPublicBuckets: true  Outputs:    BucketName:      Description: Name of the newly created S3 bucket      Value: !Ref MyS3Bucket |
| ---   MyLambdaFunction:      Type: AWS::Lambda::Function      Properties:        FunctionName: MyLambdaFunction        Handler: index.handler        Role: !GetAtt LambdaExecutionRole.Arn        Code:          S3Bucket: my-lambda-code-bucket          S3Key: lambda-function.zip        Runtime: python3.9        Timeout: 30        MemorySize: 128    LambdaExecutionRole:      Type: AWS::IAM::Role      Properties:        AssumeRolePolicyDocument:          Version: '2012-10-17'          Statement:            - Effect: Allow              Principal:                Service: lambda.amazonaws.com              Action: sts:AssumeRole        ManagedPolicyArns:          - arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole  Outputs:    RestApiId:      Description: API Gateway REST API ID      Value: !Ref MyRestApi    S3BucketName:      Description: S3 Bucket Name      Value: !Ref MyS3Bucket    LambdaFunctionArn:      Description: Lambda Function ARN      Value: !GetAtt MyLambdaFunction.Arn |

When you done, show your results to your instructor to get your sign-off.