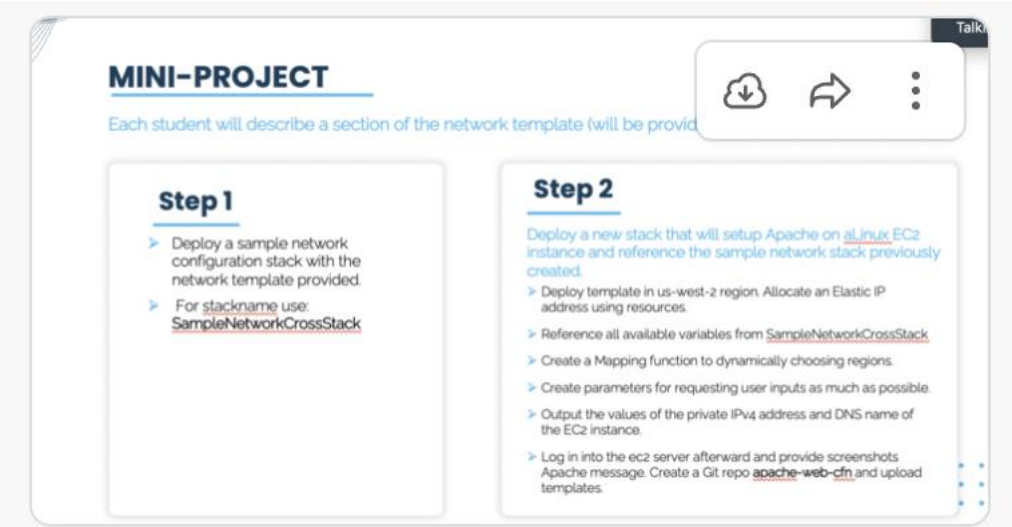
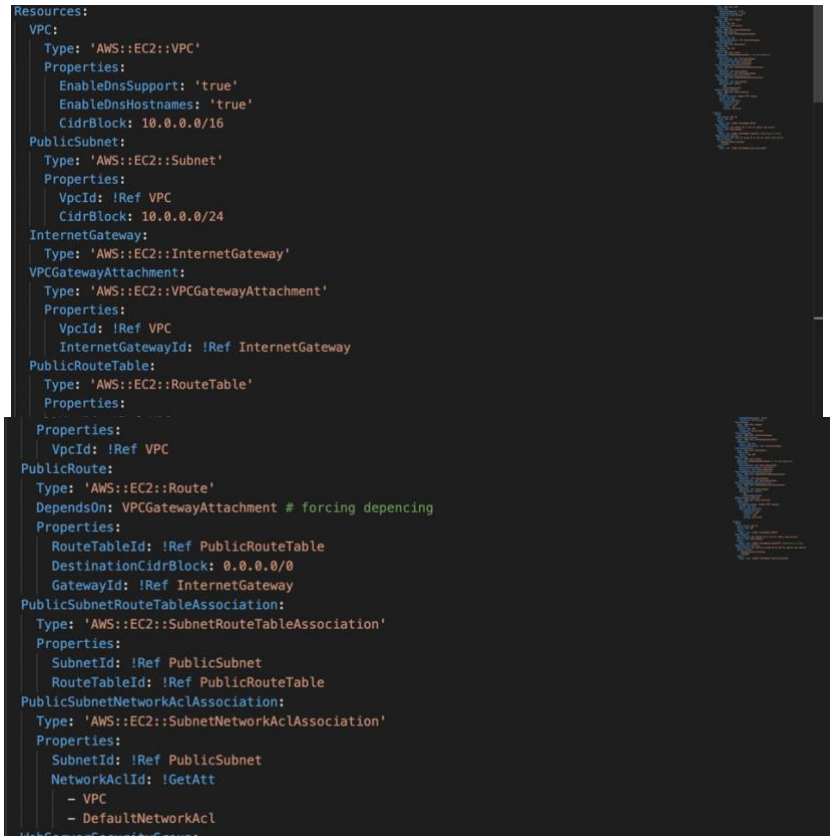
CLOUD FORMATION PROJECT

PROJECT RECOMMENDATION



SAMPLE TEMPLATE WITH EXPORT VALUES:



```
Resources:
  VPC:
    Type: 'AWS::EC2::VPC'
    Properties:
      EnableDnsSupport: 'true'
      EnableDnsHostnames: 'true'
      CidrBlock: 10.0.0.0/16
  PublicSubnet:
    Type: 'AWS::EC2::Subnet'
    Properties:
      VpcId: !Ref VPC
      CidrBlock: 10.0.0.0/24
  InternetGateway:
    Type: 'AWS::EC2::InternetGateway'
  VPCGatewayAttachment:
    Type: 'AWS::EC2::VPCGatewayAttachment'
    Properties:
      VpcId: !Ref VPC
      InternetGatewayId: !Ref InternetGateway
  PublicRouteTable:
    Type: 'AWS::EC2::RouteTable'
    Properties:
      VpcId: !Ref VPC
  PublicRoute:
    Type: 'AWS::EC2::Route'
    DependsOn: VPCGatewayAttachment # forcing depencing
    Properties:
      RouteTableId: !Ref PublicRouteTable
      DestinationCidrBlock: 0.0.0.0/0
      GatewayId: !Ref InternetGateway
  PublicSubnetRouteTableAssociation:
    Type: 'AWS::EC2::SubnetRouteTableAssociation'
    Properties:
      SubnetId: !Ref PublicSubnet
      RouteTableId: !Ref PublicRouteTable
  PublicSubnetNetworkAclAssociation:
    Type: 'AWS::EC2::SubnetNetworkAclAssociation'
    Properties:
      SubnetId: !Ref PublicSubnet
      NetworkAclId: !GetAtt
        - VPC
        - DefaultNetworkAcl
```

```
      - IpProtocol: tcp
        FromPort: '80'
        ToPort: '80'
        CidrIp: 0.0.0.0/0

Outputs:
  VPCId:
    Description: VPC ID
    Value: !Ref VPC
    Export:
      Name: !Sub '${AWS::StackName}-VPCID'
  PublicSubnet:
    Description: The subnet ID to use for public web servers
    Value: !Ref PublicSubnet
    Export:
      Name: !Sub '${AWS::StackName}-SubnetID' #replacing a string
  WebServerSecurityGroup:
    Description: The security group ID to use for public web servers
```

```
65    WebServerSecurityGroup:
66      Description: The security group ID to use for public web servers
67      Value: !GetAtt
68        - WebServerSecurityGroup
69        - GroupId
70      Export:
71        Name: !Sub '${AWS::StackName}-SecurityGroupID'
```

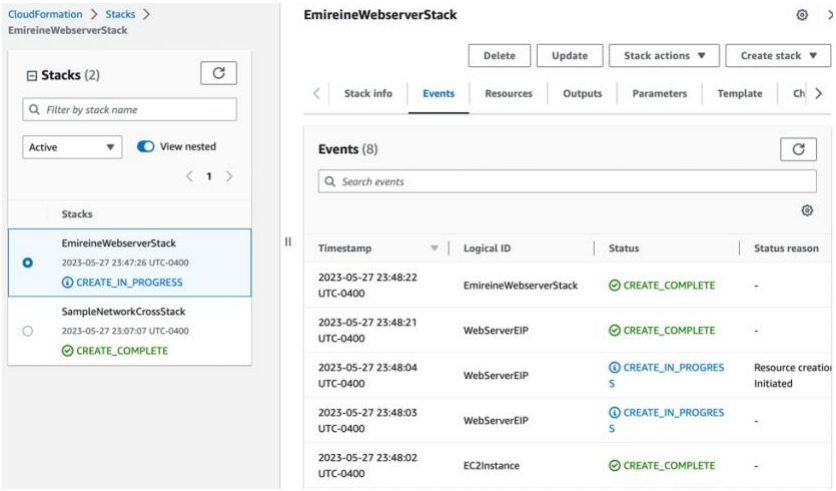
My template with import values from the sample template

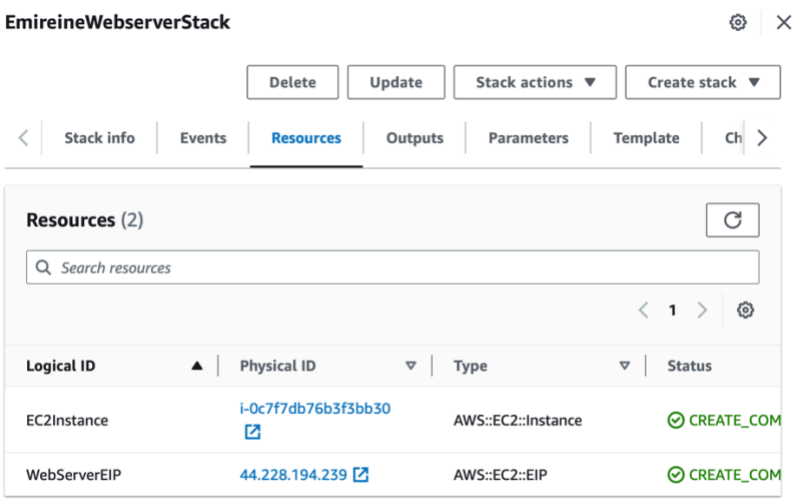
```
1   ---
2   Parameters:
3     InstanceTypeParameter:
4       Type: String
5       Description: EC2 instance type for web server
6       Default: t2.micro
7
8   Mappings:
9     EC2InstanceMappings:
10      us-west-2:
11        Key: ami-0ab193018f3e9351b
12
13  Resources:
14    EC2Instance:
15      Type: 'AWS::EC2::Instance'
16      Properties:
17        InstanceType: !Ref InstanceTypeParameter
18        ImageId: !FindInMap [EC2InstanceMappings, !Ref 'AWS::Region', Key]
19        SecurityGroupIds:
20          - "sg-023fff6b4b14e8f82"
21        SubnetId: "subnet-0d76e346bf6279ec7"
```

```
    UserData:
      Fn::Base64: !Sub |
        #!/bin/bash
        yum update -y
        yum install httpd -y
        echo '<html><body><h1 style="color:people;">welcome to CloudFormation Class as Insf
        systemctl start httpd
        systemctl enable httpd




  WebServerEIP:
    Type: AWS::EC2::EIP
    Properties:
      InstanceId: !Ref EC2Instance

Outputs:
  PrivateIPv4Address:
```

```yaml
Outputs:
  PrivateIPv4Address:
    Description: The private IPv4 address of the EC2 instance
    Value: !GetAtt EC2Instance.PrivateIp
  InstanceDNSName:
    Description: The DNS name of the EC2 instance
    Value: !GetAtt EC2Instance.PublicDnsName
```

My stack creation





Apache server successfully launched