Graduation Project DEPI_1_CAI1_ISS4_G1e AWS Cloud Solution Admin & Architect

Deploying a Highly Available Web Application with Auto Scaling using IAC

Supervised by:

Dr. Ibrahim Gomaa

Eng. Tarek Elkabani

Eng Mohamed Kababi

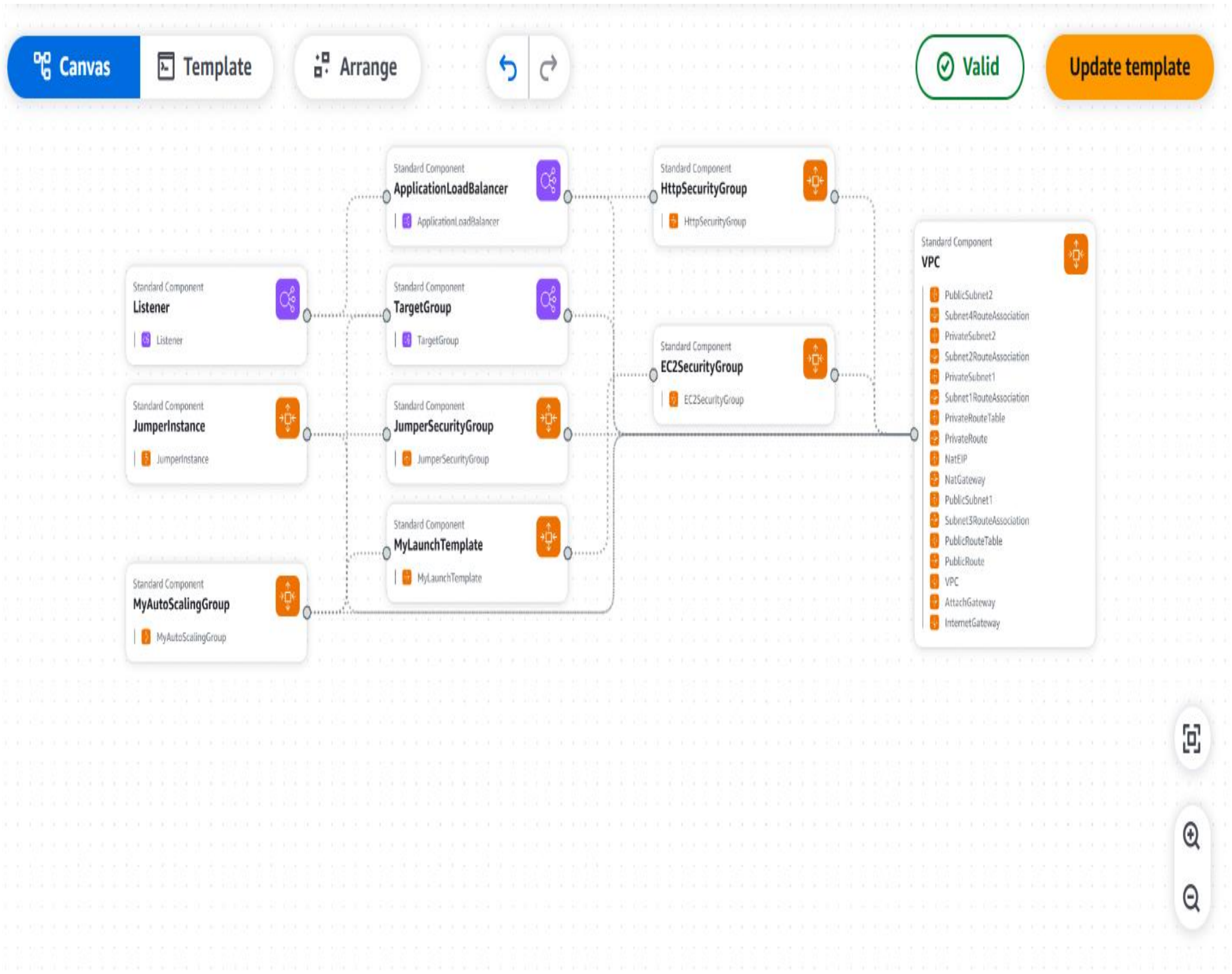
Student Name: Omar Essam Eldin Ghanim

Student Name: Amr Alaa

Student Name: Abdallah Mohamed

Student Name: Ahmed Mohsen

Student Name: Ahmed el sawi


Canvas
Template
Arrange
Valid
Update template
Standard Component
ApplicationLoadBalancer
HttpSecurityGroup
Listener
TargetGroup
EC2SecurityGroup
JumperInstance
JumperSecurityGroup
MyLaunchTemplate
MyAutoScalingGroup
VPC
PublicSubnet2
Subnet4RouteAssociation
PrivateSubnet2
Subnet2RouteAssociation
PrivateSubnet1
Subnet1RouteAssociation
PrivateRouteTable
PrivateRoute
NatEIP
NatGateway
PublicSubnet1
Subnet3RouteAssociation
PublicRouteTable
PublicRoute
VPC
AttachGateway
InternetGateway

Cloud Formation template:

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: AWS CloudFormation template for VPC, Subnets, ALB, EC2 instances,
  IAM roles, and Auto Scaling Group.

Resources:
  # VPC
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 192.168.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: DEPI-VPC

  # Subnet 1 (Private Subnet)
  PrivateSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      CidrBlock: 192.168.1.0/24
      AvailabilityZone: us-east-1a
      MapPublicIpOnLaunch: false
      Tags:
        - Key: Name
          Value: priv_subnet-1

  # Subnet 2 (Private Subnet)
  PrivateSubnet2:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      CidrBlock: 192.168.2.0/24
      AvailabilityZone: us-east-1b
      MapPublicIpOnLaunch: false
      Tags:
```

```yaml
      - Key: Name
        Value: priv_subnet-2

# Subnet 3 (Public Subnet)
PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    CidrBlock: 192.168.3.0/24
    AvailabilityZone: us-east-1a
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: pub_subnet-1

# Subnet 4 (Public Subnet)
PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    CidrBlock: 192.168.4.0/24
    AvailabilityZone: us-east-1b
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: pub_subnet-2

# Internet Gateway
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: main-gateway

AttachGateway:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    VpcId: !Ref VPC
    InternetGatewayId: !Ref InternetGateway

# Elastic IP for NAT Gateway
NatEIP:
  Type: AWS::EC2::EIP
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: nat-eip

# NAT Gateway
NatGateway:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatEIP.AllocationId
    SubnetId: !Ref PublicSubnet1
    Tags:
      - Key: Name
        Value: nat-gateway

# Route Tables
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: public_RT

PrivateRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: private_RT

# Routes
```

```yaml
PublicRoute:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PrivateRoute:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway

# Route Table Associations
Subnet1RouteAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PrivateSubnet1
    RouteTableId: !Ref PrivateRouteTable

Subnet2RouteAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PrivateSubnet2
    RouteTableId: !Ref PrivateRouteTable

Subnet3RouteAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet1
    RouteTableId: !Ref PublicRouteTable

Subnet4RouteAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet2
    RouteTableId: !Ref PublicRouteTable
```

```yaml
# Security Group for HTTP
HttpSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: HTTP Security Group
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
    SecurityGroupEgress:
      - IpProtocol: -1
        FromPort: 0
        ToPort: 0
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: HttpSecurityGroup

# Load Balancer (ALB)
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Name: ALB-DEPI
    Subnets:
      - !Ref PublicSubnet1
      - !Ref PublicSubnet2 # Public subnets for ALB
    SecurityGroups:
      - !Ref HttpSecurityGroup
    Scheme: internet-facing
    LoadBalancerAttributes:
      - Key: deletion_protection.enabled
        Value: false
    Tags:
      - Key: Name
        Value: DEPI

# Target Group
TargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: DEPI-TG
    Protocol: HTTP
    Port: 80
    VpcId: !Ref VPC
    HealthCheckPath: /
    HealthCheckProtocol: HTTP
    Tags:
      - Key: Name
        Value: DEPI

# Listener for ALB
Listener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Protocol: HTTP
    Port: 80
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref TargetGroup

# Security Group for Jumper EC2
JumperSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Jumper Security Group
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
    SecurityGroupEgress:
      - IpProtocol: -1
        FromPort: 0
        ToPort: 0
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: JumperSecurityGroup

# EC2JumpKey:
#   Type: AWS::EC2::KeyPair
#   Properties:
#     KeyName: EC2JumpKey

JumperInstance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-0866a3c8686eaeeba
    InstanceType: t2.micro
    SubnetId: !Ref PublicSubnet1
    KeyName: vockey
    SecurityGroupIds:
      - !Ref JumperSecurityGroup
    Tags:
      - Key: Name
        Value: JumpServer

# Auto Scaling Group
MyAutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  Properties:
    LaunchTemplate:
      LaunchTemplateId: !Ref MyLaunchTemplate
      Version: !GetAtt MyLaunchTemplate.LatestVersionNumber
    MinSize: 1
    MaxSize: 3
    DesiredCapacity: 1
    VPCZoneIdentifier:
      - !Ref PrivateSubnet1
```

```yaml
        - !Ref PrivateSubnet2
      TargetGroupARNs:
        - !Ref TargetGroup
      Tags:
        - Key: Name
          Value: ASG_Instance
          PropagateAtLaunch: true

  EC2SecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: EC2 Security Group
      VpcId: !Ref VPC
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
      SecurityGroupEgress:
        - IpProtocol: -1
          FromPort: 0
          ToPort: 0
          CidrIp: 0.0.0.0/0
      Tags:
        - Key: Name
          Value: EC2SecurityGroup

  MyLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: MyLaunchTemplate
      LaunchTemplateData:
        ImageId: ami-0866a3c8686eaeeba
        InstanceType: t3.micro
        # IamInstanceProfile:
```

```
        #   Name: !Ref S3InstanceProfile
        SecurityGroupIds:
          - !Ref EC2SecurityGroup
        UserData: !Base64
          Fn::Sub: |
            #!/bin/bash
            echo "Hello, World from ASG, $(hostname -f)" > /home/ec2-
user/index.html
  # S3InstanceRole:
  #   Type: AWS::IAM::Role
  #   Properties:
  #     RoleName: InstanceRole
  #     AssumeRolePolicyDocument:
  #       Version: '2012-10-17'
  #       Statement:
  #         - Effect: Allow
  #           Principal:
  #             Service:
  #               - ec2.amazonaws.com
  #           Action:
  #             - sts:AssumeRole
  #     ManagedPolicyArns:
  #       - arn:aws:iam::aws:policy/AmazonS3FullAccess

  # S3InstanceProfile:
  #   Type: AWS::IAM::InstanceProfile
  #   Properties:
  #     Path: '/'
  #     Roles:
  #       - !Ref S3InstanceRole
```