Assignment 4

1. Deploy the same node.js application in lecture 3 with AWS CLI (not EB CLI) to AWS.  Go to AWS S3, create a bucket with a name, and upload nodejs.zip to it.  Use  "S3Bucket=<replace with your bucket name>,S3Key=nodejs.zip" in your AWS CLI,  for documentation of AWS CLI,  you can refer to the link  https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environments-create-awscli.html

After deploying successfully, copy all your command to a text file and upload as solution.

aws elasticbeanstalk check-dns-availability --cname-prefix my-cname  
  
aws elasticbeanstalk describe-application-versions --application-name my-nodejs --version-label my-nodejs-v1  
  
aws elasticbeanstalk create-application --application-name my-nodejs  
  
aws elasticbeanstalk create-application-version --application-name my-nodejs --version-label my-nodejs-v1 --source-bundle S3Bucket=elasticbeanstalk-us-east-1-226439798196,S3Key=nodejs.zip  
  
aws elasticbeanstalk describe-application-versions --application-name my-nodejs --version-label my-nodejs-v1  
  
aws elasticbeanstalk create-configuration-template --application-name my-nodejs --template-name my-nodejs-v1 --solution-stack-name "64bit Amazon Linux 2 v5.8.2 running Node.js 18"  
  
aws elasticbeanstalk create-environment --cname-prefix my-cname --application-name my-nodejs --template-name my-nodejs-v1 --version-label my-nodejs-v1 --environment-name my-nodejs-v1clone --option-settings file://Desktop/git-repo/cloud-computing/assignments/options.txt  
  
aws elasticbeanstalk describe-environments --environment-names my-nodejs-v1clone

2. Deploy a CloudFormation template to create an EC2 instance named as your name, and starts up with a static IP(you can request in AWS Management Console at EC2-> Network & Security -> Elastic IPs), you can refer to the link https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/quickref-ec2.html

After deploying successfully, take a screenshot of the running EC2 instance with an static IP and your name, and upload the screenshot and your template  as solution.

A screenshot of a computer

Description automatically generated

---  
AWSTemplateFormatVersion: '2010-09-09'  
Description: 'My first CloudFormation template'  
Parameters:  
 VPC:  
 Description: 'Just select the one and only default VPC'  
 Type: 'AWS::EC2::VPC::Id'  
 Subnet:  
 Description: 'Just select one of the available subnets'  
 Type: 'AWS::EC2::Subnet::Id'  
 InstanceType:  
 Description: 'Select one of the possible instance types'  
 Type: String  
 Default: 't2.micro'  
 AllowedValues: ['t2.micro', 't2.small', 't2.medium']  
Mappings:  
 RegionMap:  
 'eu-north-1':  
 AMI: 'ami-05bc2576a72f22c39'  
 'ap-south-1':  
 AMI: 'ami-0002bdad91f793433'  
 'eu-west-3':  
 AMI: 'ami-0c73cd1c5347436f3'  
 'eu-west-2':  
 AMI: 'ami-029ed17b4ea379178'  
 'eu-west-1':  
 AMI: 'ami-04632f3cef5083854'  
 'ap-northeast-3':  
 AMI: 'ami-0ae88850834d2c589'  
 'ap-northeast-2':  
 AMI: 'ami-0263588f2531a56bd'  
 'ap-northeast-1':  
 AMI: 'ami-0abaa5b0faf689830'  
 'sa-east-1':  
 AMI: 'ami-053a035b046dbb704'  
 'ca-central-1':  
 AMI: 'ami-0173297cea9ba27b0'  
 'ap-southeast-1':  
 AMI: 'ami-0d1d4b8d5a0cd293f'  
 'ap-southeast-2':  
 AMI: 'ami-0f4484f62c4fd8767'  
 'eu-central-1':  
 AMI: 'ami-099ccc441b2ef41ec'  
 'us-east-1':  
 AMI: 'ami-061ac2e015473fbe2'  
 'us-east-2':  
 AMI: 'ami-056b1936002ca8ede'  
 'us-west-1':  
 AMI: 'ami-028f2b5ee08012131'  
 'us-west-2':  
 AMI: 'ami-0e21d4d9303512b8e'  
Resources:  
 SecurityGroup:  
 Type: 'AWS::EC2::SecurityGroup'  
 Properties:  
 GroupDescription: !Ref 'AWS::StackName'  
 VpcId: !Ref VPC  
 InstanceProfile:  
 Type: 'AWS::IAM::InstanceProfile'  
 Properties:  
 Roles:  
 - !Ref InstanceRole  
 InstanceRole:  
 Type: 'AWS::IAM::Role'  
 Properties:  
 AssumeRolePolicyDocument:  
 Version: '2012-10-17'  
 Statement:  
 - Effect: Allow  
 Principal:  
 Service: 'ec2.amazonaws.com'  
 Action: 'sts:AssumeRole'  
 ManagedPolicyArns:  
 - 'arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore'  
 EC2Instance:  
 Type: 'AWS::EC2::Instance'  
 Properties:  
 ImageId: !FindInMap [RegionMap, !Ref 'AWS::Region', AMI]  
 InstanceType: !Ref InstanceType  
 IamInstanceProfile: !Ref InstanceProfile *# instead of ec2-ssm-core we use a instance profile created in the same template for testability* SecurityGroupIds: [!Ref SecurityGroup]  
 SubnetId: !Ref Subnet  
 Tags:  
 - Key: Name  
 Value: ThanhDoNguyen  
 IPAssoc:  
 Type: AWS::EC2::EIPAssociation  
 Properties:  
 InstanceId: !Ref EC2Instance  
 EIP: 3.208.39.249  
Outputs:  
 InstanceId:  
 Value: !Ref EC2Instance  
 Description: 'Instance id (connect via Session Manager)'