Day 5: IaC

Project marks - 90/100

MCQ marks- 10/100

Deadline- 24th 9:00 AM AST

Submission link Google form with MCQ -

https://docs.google.com/forms/d/e/1FAIpQLSd_th8K5uD94stRg1iMuuudvuwR9uV4u8FmmN-5hm9dFXApHw/viewform

Submission form LMS (all days of PDF in one zip file)- learn.codingdojo.com/exams

Explanation video-

Description:

Infrastructure as Code service is gaining more traction every day as they offer distinctive benefits for businesses. Your company has hired you in an IT infrastructure management role. Your job is to provide architecture and engineering expertise to partner with your developer team to plan, analyze, design, test, and deploy infrastructure expressed as code.

You have been tasked with writing IaC templates for Cloudformation and Terraform to ensure that technical and operational requirements are achievable and being satisfied with the proposed technologies.

Please provide enough evidence of your work validating and verifying requirements, analysis and final reporting of the actual completion of the work.

in DevOps IaC services refer to a specific approach to

1. Cloudformation

    1.1. Create a cloudformation stack that will -20

        1.1.1. create a custom VPC, two ec2 servers, two security groups

        1.1.2. Bootstrap apache2 server

        1.1.3. store the application content in the index.html file printing **your name** and **cohort name**.

        1.1.4. this application should be publicly accessible.

    Paste the template yml.

Day 5: IaC

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Create a EC2 under a VPC
Resources:
  SampleVpc:
    Type: AWS::EC2::VPC
    Description: Sample VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      Tags:
        -
          Key: Name
          Value: sampleVpc
  SampleSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      CidrBlock: 10.0.0.0/24
      MapPublicIpOnLaunch: true
      VpcId: !Ref SampleVpc
  SampleRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref SampleVpc
  SampleInternetGateway:
    Type: AWS::EC2::InternetGateway
  SampleGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref SampleVpc
      InternetGatewayId: !Ref SampleInternetGateway
  InternetRoute:
    Type: AWS::EC2::Route
    DependsOn:
      - SampleGatewayAttachment
    Properties:
      RouteTableId: !Ref SampleRouteTable
      GatewayId: !Ref SampleInternetGateway
      DestinationCidrBlock: 0.0.0.0/0
  SampleSubnetRouteTableAssoc:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      RouteTableId: !Ref SampleRouteTable
      SubnetId: !Ref SampleSubnet

  SampleInstancetest1:
    Type: AWS::EC2::Instance
    DependsOn:
      - InternetRoute
      - SampleSubnetRouteTableAssoc
    Properties:
      InstanceType: t2.micro
      SubnetId: !Ref SampleSubnet
      ImageId: ami-065efef2c739d613b
      SecurityGroupIds:
        - !Ref SampleSecurityGroup
        - !Ref SampleSecurityGrouptest
      UserData:
        Fn::Base64:
          !Sub |
            #!/bin/bash
            sudo sudo
            yum update -y
            yum install httpd -y
            systemctl start httpd
            systemctl enable httpd
```

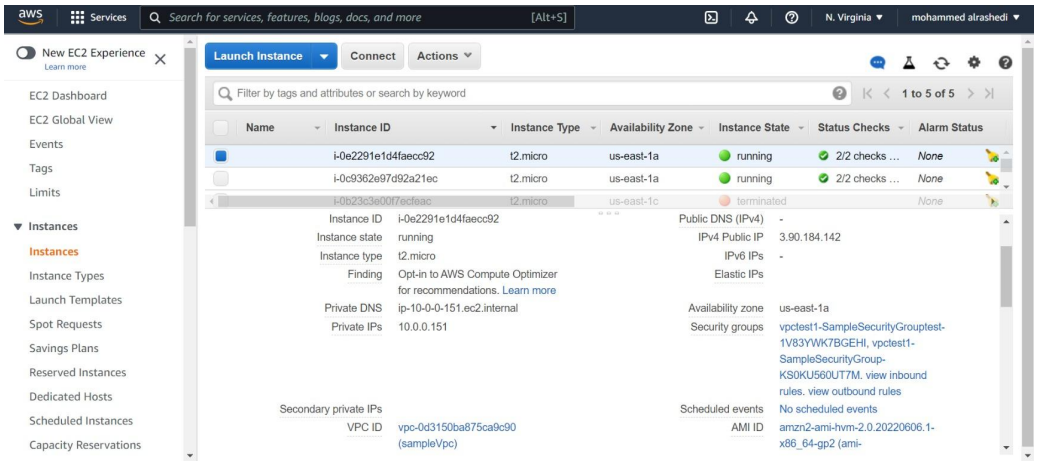
Day 5: IaC

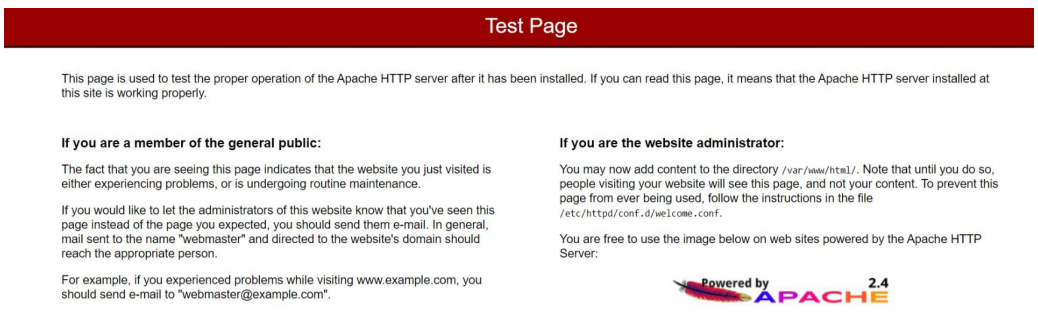
```yaml
SampleInstancetest2:
  Type: AWS::EC2::Instance
  DependsOn:
    - InternetRoute
    - SampleSubnetRouteTableAssoc
  Properties:
    InstanceType: t2.micro
    SubnetId: !Ref SampleSubnet
    ImageId: ami-065efef2c739d613b
    SecurityGroupIds:
      - !Ref SampleSecurityGroup
      - !Ref SampleSecurityGrouptest
    UserData:
      Fn::Base64:
        !Sub |
          #!/bin/bash
          sudo sudo
          yum update -y
          yum install httpd -y
          systemctl start httpd
          systemctl enable httpd
```

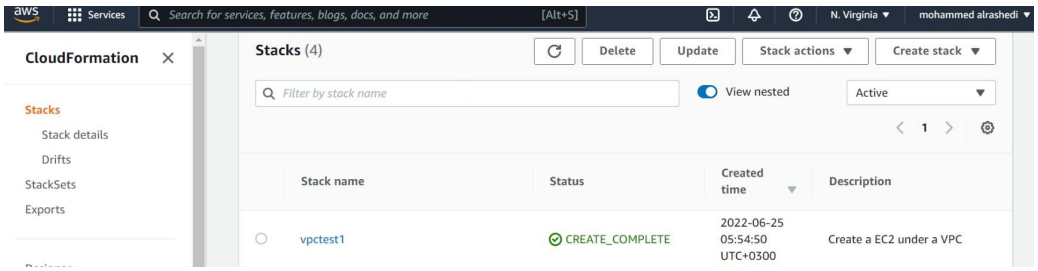
```yaml
SampleSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Sample security group
    VpcId: !Ref SampleVpc
    SecurityGroupIngress:
      -
        CidrIp: 0.0.0.0/0
        IpProtocol: tcp
        FromPort: 22
        ToPort: 22

SampleSecurityGrouptest:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Sample security group
    VpcId: !Ref SampleVpc
    SecurityGroupIngress:
      -
        CidrIp: 0.0.0.0/0
        IpProtocol: tcp
        FromPort: 80
        ToPort: 80
```

Write the commands one by one used to launch the template
aws cloudformation create-stack --stack-name vpctest1 --template-body file://vpc.yaml

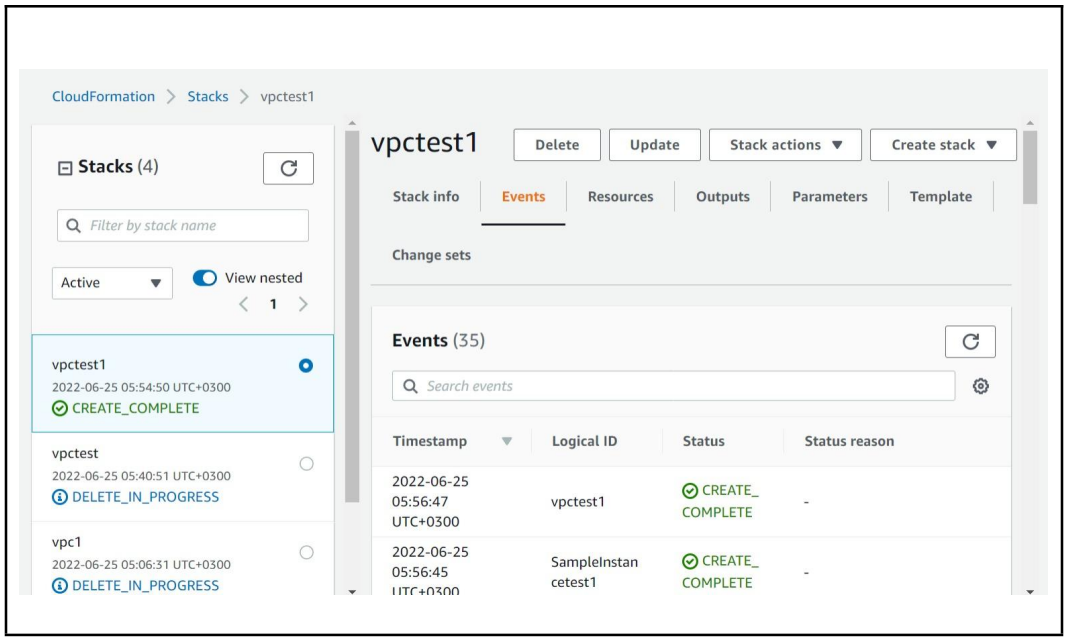
Day 5: IaC


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
mohammed alrashedi
New EC2 Experience
Learn more
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 5 of 5
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
i-0e2291e1d4faecc92 t2.micro us-east-1a running 2/2 checks ... None
i-0c9362e97d92a21ec t2.micro us-east-1a running 2/2 checks ... None
i-0b23c3e00f7ecfeac t2.micro us-east-1c terminated None
Instance ID i-0e2291e1d4faecc92
Instance state running
Instance type t2.micro
Finding Opt-in to AWS Compute Optimizer for recommendations. Learn more
Private DNS ip-10-0-0-151.ec2.internal
Private IPs 10.0.0.151
Secondary private IPs
VPC ID vpc-0d3150ba875ca9c90 (sampleVpc)
Public DNS (IPv4) -
IPv4 Public IP 3.90.184.142
IPv6 IPs -
Elastic IPs
Availability zone us-east-1a
Security groups vpctest1-SampleSecurityGrouptest-1V83YWK7BGEHI, vpctest1-SampleSecurityGroup-KS0KU560UT7M. view inbound rules. view outbound rules
Scheduled events No scheduled events
AMI ID amzn2-ami-hvm-2.0.20220606.1-x86_64-gp2 (ami-


Test Page
This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.
If you are a member of the general public:
The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.
If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.
For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".
If you are the website administrator:
You may now add content to the directory /var/www/html/. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file /etc/httpd/conf.d/welcome.conf.
You are free to use the image below on web sites powered by the Apache HTTP Server:
Powered by
APACHE
2.4


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
mohammed alrashedi
CloudFormation
Stacks
Stack details
Drifts
StackSets
Exports
Stacks (4)
Delete
Update
Stack actions
Create stack
Filter by stack name
View nested
Active
1
Stack name
Status
Created time
Description
vpctest1
CREATE_COMPLETE
2022-06-25 05:54:50 UTC+0300
Create a EC2 under a VPC

Day 5: IaC


CloudFormation > Stacks > vpctest1
Stacks (4)
Filter by stack name
Active
View nested
1
vpctest1
2022-06-25 05:54:50 UTC+0300
CREATE_COMPLETE
vpctest
2022-06-25 05:40:51 UTC+0300
DELETE_IN_PROGRESS
vpc1
2022-06-25 05:06:31 UTC+0300
DELETE_IN_PROGRESS
vpctest1
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets
Events (35)
Search events
Timestamp
Logical ID
Status
Status reason
2022-06-25 05:56:47 UTC+0300
vpctest1
CREATE_COMPLETE
-
2022-06-25 05:56:45
SampleInstancetest1
CREATE_COMPLETE
-

Day 5: IaC

1.2. Create a cloudformation with - 25

1.2.1. allowed instances 1-5

1.2.2. Default IP 0.0.0.0 and allow only standard IP format.

1.2.3. Allow only t2.nano, t2.micro, t2.small, t2.medium instances

1.2.4. Public port 80 access but port 22 access only from your own IP

1.2.5. One instances of amazon linux Latest AMI of amazon linux and one instance of ubuntu taken as parameter

1.2.6. User proper tagging with instance name - YOURNAME-cloudformation

1.2.7. Output AZ, DNS, public IP

Paste template yml.

```yaml
AWSTemplateFormatVersion: 2010-09-09
Description: Create a EC2
Parameters:
  StanderdIPFormat:
      Description : The IP address range that can be used to SSH to the EC2 instances
      Type: String
      MinLength: '9'
      MaxLength: '18'
      Default: 0.0.0.0/0
      AllowedPattern: '(\d{1,3})\.(\d{1,3})\.(\d{1,3})\.(\d{1,3})/(\d{1,2})'
      ConstraintDescription: must be a valid IP CIDR range of the form x.x.x.x/x

  InstanceTypeParameter:
    Type: String
    Default: t2.micro
    AllowedValues:
      - t2.micro
      - t2.small
      - t2.medium
      - t2.nano
    Description: Enter t2.micro, t2.small, t2.medium, or t2.nano. Default is t2.micro.
```

```yaml
Id:
  Type: AWS::EC2::Image::Id
  Default:  ami-065efef2c739d613b
  Description: Enter any AMI as Parameters
```

```yaml
Resources:
  SampleInstancetest:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType:
        Ref: InstanceTypeParameter
```

Day 5: IaC

```
      ImageId: !Ref 'Id'
      SecurityGroupIds:
        - !Ref SampleSecurityGroup
      Tags :
        - Key: name
          Value: alrashedi-cloudformation
```

```
  SampleSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Sample security group
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: !Ref StanderdIPFormat
        - IpProtocol: tcp
          FromPort: 80
          ToPort: 80
          CidrIp: 0.0.0.0/0
```

```
Outputs:
  PublicIP:
    Description: Public ip address of the newly created EC2 instance
    Value: !GetAtt [SampleInstancetest, PublicIp]
```

```
  AvailabilityZone:
   Description: AvailabilityZone of the newly created EC2 instance
   Value: !GetAtt [SampleInstancetest, AvailabilityZone]
```

```
  PublicAddress:
    Description: PublicDnsName of the newly created EC2 instance
    Value: !GetAtt [SampleInstancetest, PublicDnsName]
```

Write the commands one by one used to launch the template :

Aws cloudformation create-stack --stack-name mohammed --template-body File://ec2.yamlparameters parameterKey=instanceTypeparameter.parameterValue=t2.nano

Day 5: IaC

2. Terraform

2.1. Dojo jump is going to be launched soon. It aims to deploy it in Apache Servers. You and your colleagues have started to work on the project. Your Teammate have developed the website and they need your help to build infrastructure for deploying the website. - 20

Dojo-jump game link - https://github.com/chandradeoarya/dojo-jump

2.1.1. Ubuntu or amazon linux in server

2.1.2. Proper ingress and egress

2.1.3. Proper tagging

Paste all the template yml files like output, variable, main etc.

Day 5: IaC

Install.sh

```
#!/bin/sh
sudo su
yum update -y
yum install -y httpd.x86_64
chmod -R 777 /var/www/html
cd /var/www/html
wget https://raw.githubusercontent.com/chandradeoarya/dojo-jump/master/style.css
wget https://raw.githubusercontent.com/chandradeoarya/dojo-jump/master/main.js
wget https://raw.githubusercontent.com/chandradeoarya/dojo-
jump/master/index.html
systemctl start httpd.service
systemctl enable httpd.service
```

Main.tf

```
resource "aws_key_pair" "default" {
  key_name   = "key"
  public_key = file("${var.key_path}")
}

# Define the security group
resource "aws_security_group" "sgweb" {
  name        = "DojoJump-ml"
  description = "Allow incoming HTTP connections & SSH access"

  ingress {
    description = "Allow incoming HTTP connections"
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    description = "Allow incoming SSH access"
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
```

```
  tags = {
    Name = "DojoJump mo"
  }
}

# Define an Amazon Linux instance with Apache web server
resource "aws_instance" "DojoJump" {
  ami                    = var.amazon_linux_ami
  instance_type          = var.instance_type
  key_name               = aws_key_pair.default.id
  vpc_security_group_ids = ["${aws_security_group.sgweb.id}"]
  user_data              = file("install.sh")

  tags = {
    Name = "DojoJumpmo"
  }

}
```

Provider.tf

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "4.18.0"
    }
  }
}

# Define AWS as a provider
provider "aws" {
  region = var.aws_region
}
```

 ##

Variables.tf

```
variable "aws_region" {
  description = "Region for the EC2"
  default     = "us-east-1"
}

variable "amazon_linux_ami" {
  description = "Amazon linux AMI for EC2"
  default     = "ami-0cff7528ff583bf9a"
}

variable "instance_type" {
  description = "instance type"
  default     = "t2.micro"
}
```

Day 5: IaC

```
variable "key_path" {
  description = "SSH Public Key path"
  default     = "key.pub"
}
```

Output.tf

```
output "instance_id_DojoJump" {
  description = "Instance ID"
  value       = aws_instance.DojoJump.id
}

output "instance_public_ip_DojoJump" {
  description = "Instance Public IP"
  value       = aws_instance.DojoJump.public_ip
}

output "instance_AZ_name_DojoJump" {
  description = "availability zone"
  value       = aws_instance.DojoJump.availability_zone
}
```

Day 5: IaC

Write the commands one by one used to launch the template .
Terraform init
Terraform plan
Terraform apply

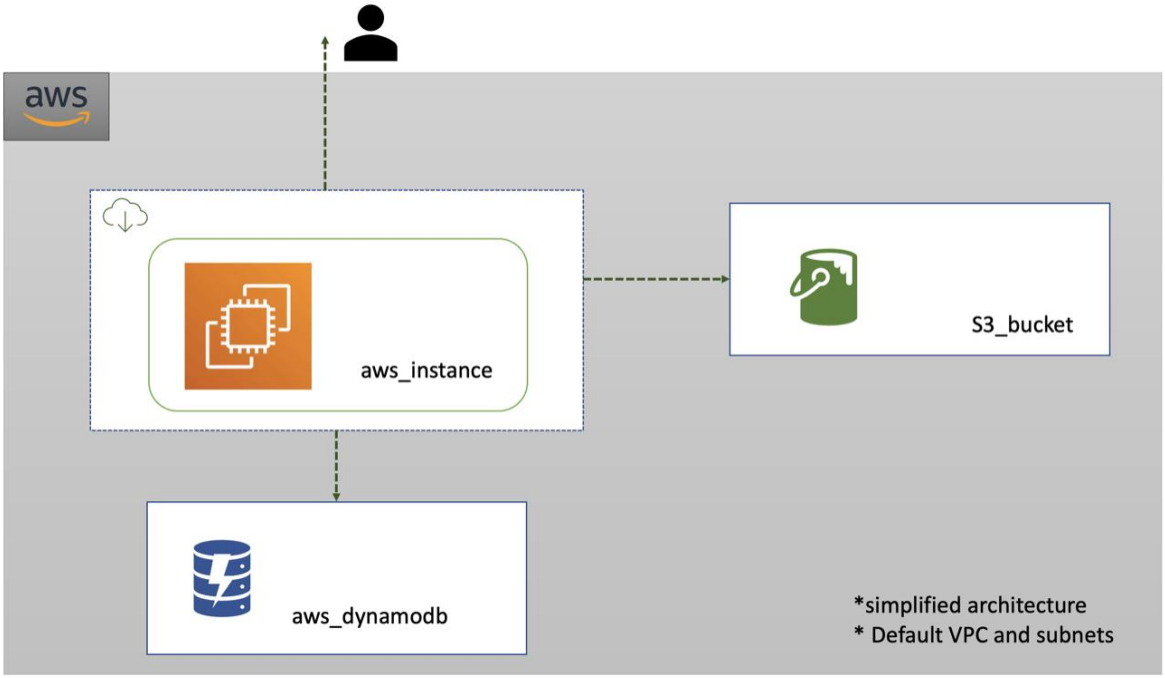
2.2. Terraform custom module - 25

Let us consider that an organization “CodingDojo” has a blueprint of a prototype of an application “Payroll” that needs to deploy at several countries.

Each country will have its own instance of software deployed on an AWS instance using the same architecture.

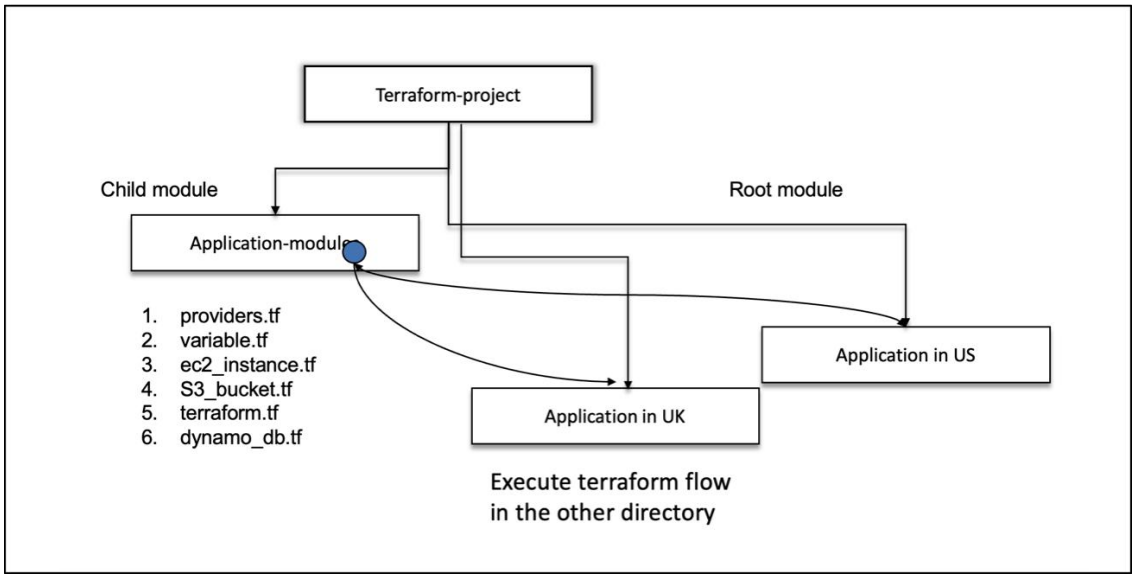
Architecture design –

Day 5: IaC



Important checkpoints:

1. It consists of a single ec2instance using the custom AMI, that holds the application server.
2. A DynamoDB NoSQL database that will be used to store the data of the employees
3. S3 bucket which will be used to save tax and other documents.
4. Users can access the application through this EC2 instance.
5. This is the architecture of the application in most simplified form.
6. Default VPC and subnets are used.



Day 5: IaC

Create a custom module to deploy the whole application infrastructure.

Use this module to deploy the application in us-east-1. Paste the screenshot of the resources created.

Paste all the template yml files like output, variable, main etc.

Day 5: IaC

Write the commands one by one used to launch the template.