Midterm Report– Spring 2024 Topic: CDK for Terraform

Name: Wendimu Dinsa

What is CDK for Terraform

1.What is CDK for Terraform

- ✓CDK (Cloud Development Kit) for Terraform is a software framework for defining cloud infrastructure as a code using familiar programming languages.
- ✓It allows developers to define infrastructure using high-level construct and abstraction, making it easier to manage and maintain infrastructure code.

1.1.Feature of CDK for Terraform

- ✓Familiar programming languages
- ✓Higher-level constructs
- ✓Type safety and IDE support.
- ✓Code reuse and sharing.
- ✓Integration with Terraform ecosystem.

Static Website on S3

Step-by-Step Instructions

What is different from our classroom lab?

2. To create a Static Website on Amazon S3 using Infrastructure as a Code (IaC) we use AWS CloudFormation, AWS CDK (Cloud Development Kit), or Terraform. Here below are the step-by-step instructions using AWS CloudFormation.

✓Create a CloudFormation template in JSON or YAML format.

Example

```
{
  "Resources": {
    "S3Bucket": {
      "Type": "AWS::S3::Bucket",
      "Properties": {
        "AccessControl": "PublicRead",
        "WebsiteConfiguration": {
          "IndexDocument": "index.html",
          "ErrorDocument": "error.html"
        }
```

```
      }
    }
  }
}
```

- ✓ Upload Website files to S3

  Place your website in local directory

- ✓ Deploy CloudFormation Stack

  Using AWS Management Console, AWS CLI, Or to Deploy the CloudFormation stack.

- ✓ Make Files Public: After your stack is created, make the files public by setting the backet policy

Example

```
{
  "Type": "AWS::S3::BucketPolicy",
  "Properties": {
    "Bucket": {
      "Ref": "S3Bucket"
```

```
    },
    "PolicyDocument": {
      "Statement": [
        {
          "Sid": "PublicReadGetObject",
          "Effect": "Allow",
          "Principal": "*",
          "Action": "s3:GetObject",
          "Resource": {
            "Fn::Sub": "arn:aws:s3:::${S3Bucket}/*"
          }
        }
      ]
    }
  }
}
```

✓At the end Access your website

Once the Stack Creation is complete, you can access your website on Amazon S3 using IaC

What is different from our classroom lab?

There is no difference in how it works but I take different names for words

What's the benefit of using CDK for Terraform?

There are several benefits of using CDK to generate Terraform configurations. These are:

- High-level Constructs
- Leveraging Programming languages Type safety and IDE support
- Reuse and modularity.
- Integration with AWS constructs
- Cloud-native abstractions
- Ecosystem and Community

Tutorial 2: Website on EC2 instance

Step-by-step Instructions

- First, prepare your project structure.
  - Create a directory for your project
  - Inside the directory create a cloud formation template file
- Write a cloud formation Template.

  Example

```
{
  "Resources": {
    "WebServerInstance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "your-ami-id",
        "InstanceType": "t2.micro",
        "KeyName": "your-key-pair",
        "SecurityGroups": [
```

```
          { "Ref": "WebServerSecurityGroup" }
        ],
        "UserData": {
          "Fn::Base64": {
            "Fn::Sub": "#!/bin/bash\nyum update -
y\nyum install -y httpd\nsystemctl start
httpd\nsystemctl enable httpd\n"
          }
        }
      }
    },
    "WebServerSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Allow HTTP access",
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIp": "0.0.0.0/0"
          }
        ]
```

```
    }
   }
  }
}
```

Steps

- Upload website files to S3
- Update CloudFormation Template
- Deploy Stack
- Wait for stack deployments.
- Access your EC2 Instance
- Test your website.
- Update and clean up

What is different from our classroom lab?

The difference between the classroom and this is that; in the classroom lab the by-step is discussed in code and this one is the general step in type

CloudFormation vs. Terraform o CDK (for CloudFormation) vs CDK for Terraform

Terraform VS CloudFormation Both are IaC tools to provision and manage cloud resources	
Terraform	**CloudFormation**
✓ Developed by HashiCorp, It supports multiple cloud providers such	✓ Developed and maintained by AWS, and tightly integrated with AWS services.

as AWS, Azure, Google Cloud Platform, and many others. It provides consistent workflow across different providers.	
✓It uses its declarative configuration language called HashiCorp Configuration Language (HCL). HCL is often Considered more readable and expressive compared to JSON and YAML	✓It uses JSON and YAML to define infrastructure resources in templates
✓Maintains State file that keeps track of the infrastructure it manages. This allows for features like	✓automatically manages the state of resources being provisioned but this state is not easily

resource dependency tracking, planning, and applying operations, and collaboration among team members	accessible or manageable by users
✓ Supports a vast number of resources across various cloud providers, and its modular design allows for community-contributed modules to extend its capabilities.	✓ Provides a wide range of AWS resource types and features but may lag in supporting newer AWS services or features.
✓ Has a large and active community, with a rich ecosystem of third-party providers, modules, and integrations beyond	✓ Being tightly integrated with AWS, it benefits from AWS's ecosystem and official support.

AWS.	
✓It's an open-source tool, so there are no direct costs associated with using Terraform. However, there might be costs associated with the resources it provides.	✓Generally, there are no additional costs for using CloudFormation beyond the resources it provisions.

CDK (for CloudFormation Vs for Terraform)

CDK (for CloudFormation Vs CDK for Terraform)	
CDK for CloudFormation	**CDK for Terraform**
Language Support ✓Allows developers to define AWS infrastructure using programming	***Language Support*** ✓Allows developers to define infrastructure using programming languages such as

languages such as Java, Python, and others. ***Integration*** ✓ Generates CloudFormation Template behind the scenes. ***Ecosystem*** ✓ Official support, documentation, and integration with AWS services	Java, Python, and others and generates configuration files. ***Integration*** ✓ Bridges the gap between Terraform Declarative language and more expressive programming language syntax. ***Ecosystem*** ✓ CDK for terraform extends its ecosystem of providers and modules by allowing developers to leverage the ecosystem including libraries, framework, and development tools

Reference:

https://chat.openai.com/c/fd6e6648-8fe1-4b6f-b5ac-c2e8403da1bf

https://chat.openai.com/c/341092cf-66a2-4436-90f2-dc0b9166f0a7

https://www.slideshare.net/ServNet/cloudformation-vs-terraform-vs-ansible

https://www.toptal.com/terraform/terraform-vs-cloudformation#:~:text=Terraform%20and%20CloudFormation%20are%20both,wide%20range%20of%20cloud%20vendors.