AWS Coaching

Amazon AWS Cloud Development Kit (CDK)

Jeroen van Ooststroom

Amazon AWS Cloud Development Kit (CDK)

What is it?

The AWS Cloud Development Kit (AWS CDK) is an open-source software development framework for defining cloud infrastructure in code and provisioning it through AWS CloudFormation.

The AWS CDK consists of two primary parts:

- AWS CDK Construct Library
  (TypeScript, JavaScript, Python, Java, C#, Go)
- AWS CDK Command Line Interface (AWS CDK CLI)

Amazon AWS Cloud Development Kit (CDK)

Level 1 (L1) Constructs

- Also known as *CFN resources*
- Lowest-level construct
- Offer no abstraction
- Maps directly to a single AWS CloudFormation resource
- Offer complete control over defining AWS resource properties

Amazon AWS Cloud Development Kit (CDK)

Level 2 (L2) Constructs

- Also known as *curated* constructs
- Thoughtfully developed by the CDK team
- Provide higher-level of abstraction
- Map directly to single AWS CloudFormation resources
- Include sensible default property configurations, best practice security policies, and generate a lot of boilerplate and glue logic

Amazon AWS Cloud Development Kit (CDK)

Level 3 (L3) Constructs

- Also know as *patterns*
- Provide highest-level of abstraction
- Contain a collection of resources to accomplish a specific task or service
- Used to create entire AWS architectures for particular use cases

Amazon AWS Cloud Development Kit (CDK)

Prerequisites

- Node.js:
  - Mac OS (brew): ‘`brew install node`’
  - Windows Installer: https://nodejs.org/en/download

Amazon AWS Cloud Development Kit (CDK)

Installation

- Use one of the following 2 options:
  - `npm install -g aws-cdk`
  - `npm install -g aws-cdk@X.YY.Z` *(i.e.: `2.189.0`)*
- Deploy the CDK toolkit into an AWS environment:
  - `cdk bootstrap`

Amazon AWS Cloud Development Kit (CDK)

Create CDK project

- Use the following command to create your CDK project:
  - `cdk init app --language java`

Amazon AWS Cloud Development Kit (CDK)

AWS CDK & CLI links

- AWS CDK:
  - What is the AWS CDK?

    https://docs.aws.amazon.com/cdk/v2/guide/home.html
  - AWS CDK CLI command reference

    https://docs.aws.amazon.com/cdk/v2/guide/ref-cli-cmd.html
- AWS CLI: https://docs.aws.amazon.com/cli/latest/

Amazon AWS Cloud Development Kit (CDK)

Describe old VPCs *(using AWS CLI)*

```
aws ec2 describe-vpcs --vpc-ids vpc-0d8e93eb55a0c2180
```

```
{
    "Vpcs": [
        {
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-03b568cf248dd45b3",
                    "CidrBlock": "10.0.0.0/24",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-0d8e93eb55a0c2180",
            "State": "available",
            "CidrBlock": "10.0.0.0/24",
            "DhcpOptionsId": "dopt-02a939ad6a846c172"
        }
    ]
}
```

Amazon AWS Cloud Development Kit (CDK)

Describe VPCs *(using AWS CLI)*

```
aws ec2 describe-vpcs --vpc-ids vpc-0d8e93eb55a0c2180 --output table
```

```
-----------------------------------------------------------------------------------------------------------------------------------
|                                                          DescribeVpcs                                                           |
+---------------------------------------------------------------------------------------------------------------------------------+
||                                                             Vpcs                                                              ||
|+-------------+-------------------------+------------------+------------+---------------+-------------+-------------------------+|
||  CidrBlock  |      DhcpOptionsId      | InstanceTenancy  | IsDefault  |    OwnerId    |    State    |          VpcId          ||
|+-------------+-------------------------+------------------+------------+---------------+-------------+-------------------------+|
||  10.0.0.0/24|  dopt-02a939ad6a846c172 |  default         |  False     |  039612879714 |  available  |  vpc-0d8e93eb55a0c2180  ||
|+-------------+-------------------------+------------------+------------+---------------+-------------+-------------------------+|
|||                                                   BlockPublicAccessStates                                                   |||
||+----------------------------------------------------------------------------------------------------+------------------------+||
|||  InternetGatewayBlockMode                                                                          |  off                   |||
||+----------------------------------------------------------------------------------------------------+------------------------+||
|||                                                   CidrBlockAssociationSet                                                   |||
||+----------------------------------------------------------------------------------------+------------------------------------+||
|||                                      AssociationId                                     |             CidrBlock              |||
||+----------------------------------------------------------------------------------------+------------------------------------+||
|||  vpc-cidr-assoc-03b568cf248dd45b3                                                      |  10.0.0.0/24                       |||
||+----------------------------------------------------------------------------------------+------------------------------------+||
||||                                                      CidrBlockState                                                       ||||
|||+-----------------------------------------------+---------------------------------------------------------------------------+|||
||||  State                                        |  associated                                                               ||||
|||+-----------------------------------------------+---------------------------------------------------------------------------+|||
```

Amazon AWS Cloud Development Kit (CDK)

Create a VPC *(using L1 construct)*

```java
public class IacStack
extends Stack {
    private static final Logger LOGGER = Logger.getLogger(IacStack.class.getName());

    private static final String PREFIX = "jeroens-";

    public IacStack(final Construct scope, final String id) {
        this(scope, id, null);
    }

    public IacStack(final Construct scope, final String id, final StackProps props) {
        super(scope, id, props);

        CfnOutput.Builder.create(this, "AccountUsed").
            description("").
            value("Account: " + this.getAccount()).
            build();
        CfnOutput.Builder.create(this, "RegionUsed").
            description("").
            value("Region: " + this.getRegion()).
            build();

        var vpc = createVpc("10.0.0.0/16", "vpc");
    }

    private CfnVPC createVpc(final String cidrBlock) {
        var vpc = CfnVPC.Builder.create(this, PREFIX + "vpc").cidrBlock(cidrBlock).build();
        Tags.of(vpc).add("Name", PREFIX + "vpc");
        CfnOutput.Builder.create(this, "VpcCreated").value("VpcId: " + vpc.getAttrVpcId()).build();
        return vpc;
    }
}
```

Amazon AWS Cloud Development Kit (CDK)

Synthesize & Deploy

- To synthesize and print the CloudFormation template for the stack use the following command:

  `cdk synth` *(Please note: This is not a mandatory step.)*

- To deploy the stack into the AWS account use the following command:

  ```
  cdk deploy
  ```

Amazon AWS Cloud Development Kit (CDK)

Describe new VPCs *(using AWS CLI)*

```
aws ec2 describe-vpcs --vpc-ids vpc-0c3642cfd52cdc5ec
```

```
{
    "Vpcs": [
        {
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0febdee48a0db600d",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-0c3642cfd52cdc5ec",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-02a939ad6a846c172"
        }
    ]
}
```

Amazon AWS Cloud Development Kit (CDK)

Describe old public Subnets *(using AWS CLI)*

```
aws ec2 describe-subnets --filters "Name=tag:Name,Values=jdriven-subnet-public1-eu-central-1a"
```

```
{
    "Subnets": [
        {
            "AvailabilityZoneId": "euc1-az2",
            "MapCustomerOwnedIpOnLaunch": false,
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-central-1:039612879714:subnet/subnet-0289178bb8510fa8c",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0289178bb8510fa8c",
            "State": "available",
            "VpcId": "vpc-0d8e93eb55a0c2180",
            "CidrBlock": "10.0.0.0/28",
            "AvailableIpAddressCount": 10,
            "AvailabilityZone": "eu-central-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
```

Amazon AWS Cloud Development Kit (CDK)

Create a public Subnet *(using L2 construct)*

```java
public IacStack(final Construct scope, final String id, final StackProps props) {
    ...

    var vpc = createVpc("10.0.0.0/16");
    var publicSubnet = createPublicSubnet("10.0.101.0/24", vpc.getAttrVpcId());
}

private ISubnet createPublicSubnet(final String cidrBlock, final String vpcId) {
    var publicCfnSubnet =
        CfnSubnet.Builder.create(this, PREFIX + "public-cfn-subnet").
            availabilityZone(getRegion() + "a").
            cidrBlock(cidrBlock).
            mapPublicIpOnLaunch(true).
            vpcId(vpcId).
            build();
    Tags.of(publicCfnSubnet).add("Name", PREFIX + "public-subnet");
    var publicSubnet =
        Subnet.fromSubnetId(this, PREFIX + "public-subnet", publicCfnSubnet.getAttrSubnetId());
    CfnOutput.Builder.create(this, "PublicSubnetCreated").
        value("SubnetId: " + publicSubnet.getSubnetId()).build();
    return publicSubnet;
}
```

Amazon AWS Cloud Development Kit (CDK)

Describe new public Subnets *(using AWS CLI)*

```
aws ec2 describe-subnets --filters "Name=tag:Name,Values=jeroens-public-subnet"
```

```
{
    "Subnets": [
        {
            "AvailabilityZoneId": "euc1-az2",
            "MapCustomerOwnedIpOnLaunch": false,
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-central-1:039612879714:subnet/subnet-05436707d1457420a",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-05436707d1457420a",
            "State": "available",
            "VpcId": "vpc-0c3642cfd52cdc5ec",
            "CidrBlock": "10.0.101.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "eu-central-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": true
        }
    ]
}
```

Amazon AWS Cloud Development Kit (CDK)

Describe old private Subnets *(using AWS CLI)*

```
aws ec2 describe-subnets --filters "Name=tag:Name,Values=jdriven-subnet-private1-eu-central-1a"
```

```
{
    "Subnets": [
        {
            "AvailabilityZoneId": "euc1-az2",
            "MapCustomerOwnedIpOnLaunch": false,
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-central-1:039612879714:subnet/subnet-094ca854e990311c2",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-094ca854e990311c2",
            "State": "available",
            "VpcId": "vpc-0d8e93eb55a0c2180",
            "CidrBlock": "10.0.0.128/28",
            "AvailableIpAddressCount": 9,
            "AvailabilityZone": "eu-central-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
```

Amazon AWS Cloud Development Kit (CDK)

Create a private Subnet *(using L2 construct)*

```
public IacStack(final Construct scope, final String id, final StackProps props) {
    ...

    var vpc = createVpc("10.0.0.0/16");
    var publicSubnet = createPublicSubnet("10.0.101.0/24", vpc.getAttrVpcId());
    var privateSubnet = createPrivateSubnet("10.0.201.0/24", vpc.getAttrVpcId());
}

private ISubnet createPrivateSubnet(final String cidrBlock, final String vpcId) {
    var privateCfnSubnet =
        CfnSubnet.Builder.create(this, PREFIX + "private-cfn-subnet").
            availabilityZone(getRegion() + "a").
            cidrBlock(cidrBlock).
            vpcId(vpcId).
            build();
    Tags.of(privateCfnSubnet).add("Name", PREFIX + "private-subnet");
    var privateSubnet =
        Subnet.fromSubnetId(this, PREFIX + "private-subnet", privateCfnSubnet.getAttrSubnetId());
    CfnOutput.Builder.create(this, "PrivateSubnetCreated").
        value("SubnetId: " + privateSubnet.getSubnetId()).build();
    return privateSubnet;
}
```

Amazon AWS Cloud Development Kit (CDK)

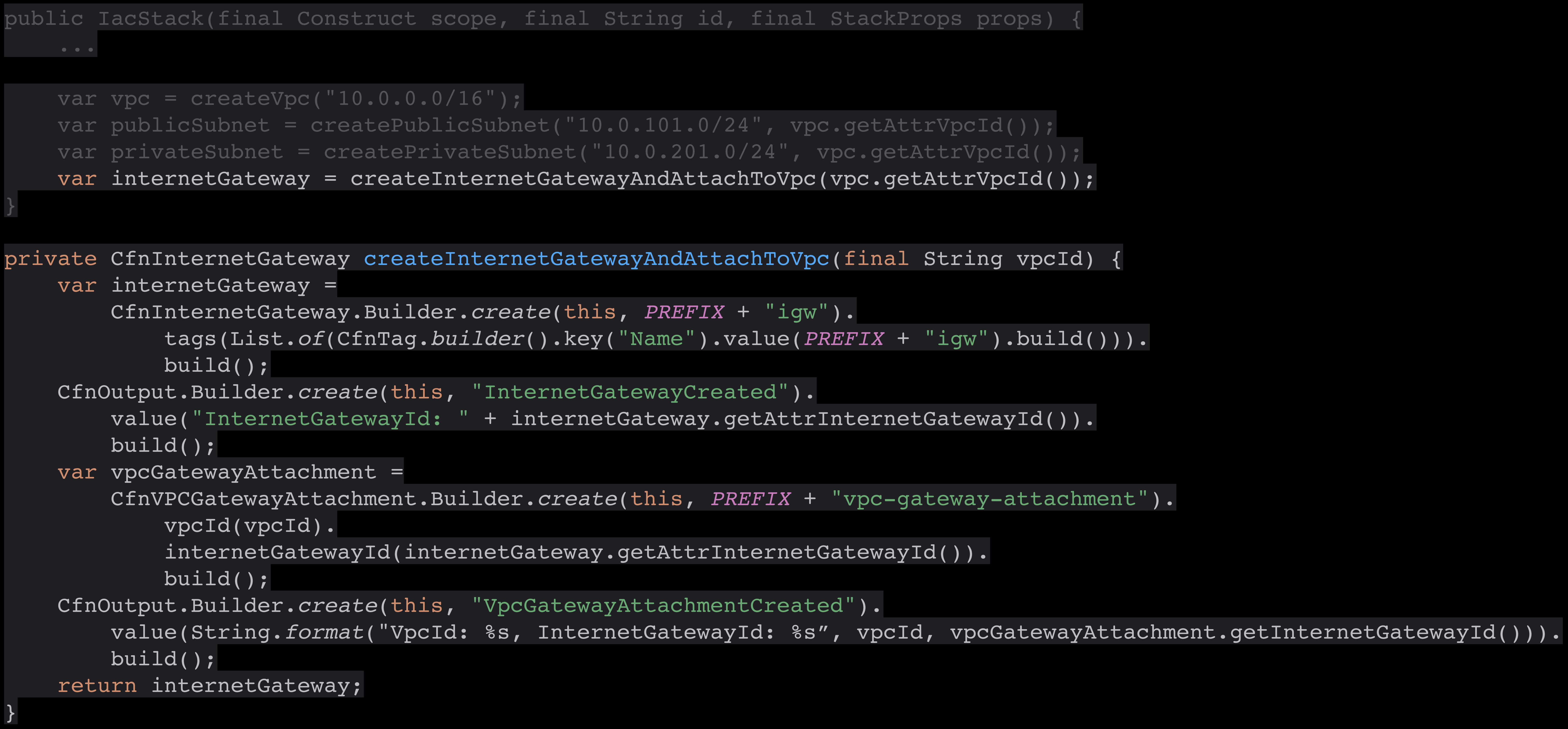
Describe new private Subnets *(using AWS CLI)*

```
aws ec2 describe-subnets --filters "Name=tag:Name,Values=jeroens-private-subnet"
```

```
{
    "Subnets": [
        {
            "AvailabilityZoneId": "euc1-az2",
            "MapCustomerOwnedIpOnLaunch": false,
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-central-1:039612879714:subnet/subnet-0cb9e46be6ec586b5",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0cb9e46be6ec586b5",
            "State": "available",
            "VpcId": "vpc-0c3642cfd52cdc5ec",
            "CidrBlock": "10.0.201.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "eu-central-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
```

Amazon AWS Cloud Development Kit (CDK)

Create an Internet Gateway



```
public IacStack(final Construct scope, final String id, final StackProps props) {
    ...

    var vpc = createVpc("10.0.0.0/16");
    var publicSubnet = createPublicSubnet("10.0.101.0/24", vpc.getAttrVpcId());
    var privateSubnet = createPrivateSubnet("10.0.201.0/24", vpc.getAttrVpcId());
    var internetGateway = createInternetGatewayAndAttachToVpc(vpc.getAttrVpcId());
}

private CfnInternetGateway createInternetGatewayAndAttachToVpc(final String vpcId) {
    var internetGateway =
        CfnInternetGateway.Builder.create(this, PREFIX + "igw").
            tags(List.of(CfnTag.builder().key("Name").value(PREFIX + "igw").build())).
            build();
    CfnOutput.Builder.create(this, "InternetGatewayCreated").
        value("InternetGatewayId: " + internetGateway.getAttrInternetGatewayId()).
        build();
    var vpcGatewayAttachment =
        CfnVPCGatewayAttachment.Builder.create(this, PREFIX + "vpc-gateway-attachment").
            vpcId(vpcId).
            internetGatewayId(internetGateway.getAttrInternetGatewayId()).
            build();
    CfnOutput.Builder.create(this, "VpcGatewayAttachmentCreated").
        value(String.format("VpcId: %s, InternetGatewayId: %s”, vpcId, vpcGatewayAttachment.getInternetGatewayId())).
        build();
    return internetGateway;
}
```

Amazon AWS Cloud Development Kit (CDK)

Create a public Route Table *(Step 1)*

```java
public IacStack(final Construct scope, final String id, final StackProps props) {
    ...

    var vpc = createVpc("10.0.0.0/16");
    var publicSubnet = createPublicSubnet("10.0.101.0/24", vpc.getAttrVpcId());
    var privateSubnet = createPrivateSubnet("10.0.201.0/24", vpc.getAttrVpcId());
    var internetGateway = createInternetGatewayAndAttachToVpc(vpc.getAttrVpcId());
    var publicRouteTable = createPublicRouteTable(vpc, internetGateway, publicSubnet);
}

private CfnRouteTable createPublicRouteTable(
        final CfnVPC vpc, final CfnInternetGateway internetGateway, final ISubnet subnet) {

    var publicRouteTable =
            CfnRouteTable.Builder.create(this, PREFIX + "public-route-table").
                    vpcId(vpc.getAttrVpcId()).
                    tags(List.of(CfnTag.builder().key("Name").value("public-route-table").build())).
                    build();
    CfnOutput.Builder.create(this, PREFIX + "PublicRouteTableCreated").
            value("RouteTableId: " + publicRouteTable.getAttrRouteTableId()).
            build();
    return publicRouteTable;
}
```

Amazon AWS Cloud Development Kit (CDK)

Create a public Route Table *(Step 2)*

```
public IacStack(final Construct scope, final String id, final StackProps props) {
    ...

    var vpc = createVpc("10.0.0.0/16");
    var publicSubnet = createPublicSubnet("10.0.101.0/24", vpc.getAttrVpcId());
    var privateSubnet = createPrivateSubnet("10.0.201.0/24", vpc.getAttrVpcId());
    var internetGateway = createInternetGatewayAndAttachToVpc(vpc.getAttrVpcId());
    var publicRouteTable = createPublicRouteTable(vpc, internetGateway, publicSubnet);
}

private CfnRouteTable createPublicRouteTable(
        final CfnVPC vpc, final CfnInternetGateway internetGateway, final ISubnet subnet) {

    var publicRouteTable =
            CfnRouteTable.Builder.create(this, PREFIX + "public-route-table").
                    vpcId(vpc.getAttrVpcId()).
                    tags(List.of(CfnTag.builder().key("Name").value("public-route-table").build())).
                    build();
    var localRoute =
            CfnRoute.Builder.create(this, PREFIX + "local-route").
                    routeTableId(publicRouteTable.getAttrRouteTableId()).
                    destinationCidrBlock(vpc.getAttrCidrBlock()).
                    gatewayId("local").
                    build();
    var internetGatewayRoute =
            CfnRoute.Builder.create(this, PREFIX + “internet-gateway-route").
                    routeTableId(publicRouteTable.getAttrRouteTableId()).
                    destinationCidrBlock("0.0.0.0/0").
                    gatewayId(internetGateway.getAttrInternetGatewayId()).
                    build();
    ...
    return publicRouteTable;
}
```

Amazon AWS Cloud Development Kit (CDK)

Create a public Route Table *(Step 3)*

```
public IacStack(final Construct scope, final String id, final StackProps props) {
    ...

    var vpc = createVpc("10.0.0.0/16");
    var publicSubnet = createPublicSubnet("10.0.101.0/24", vpc.getAttrVpcId());
    var privateSubnet = createPrivateSubnet("10.0.201.0/24", vpc.getAttrVpcId());
    var internetGateway = createInternetGatewayAndAttachToVpc(vpc.getAttrVpcId());
    var publicRouteTable = createPublicRouteTable(vpc, internetGateway, publicSubnet);
}

private CfnRouteTable createPublicRouteTable(
        final CfnVPC vpc, final CfnInternetGateway internetGateway, final ISubnet subnet) {

    ...
    var subnetRouteTableAssociation =
            CfnSubnetRouteTableAssociation.Builder.create(this, PREFIX + "subnet-route-table-association").
                    subnetId(subnet.getSubnetId()).
                    routeTableId(publicRouteTable.getAttrRouteTableId()).
                    build();
    CfnOutput.Builder.create(this, PREFIX + "SubnetRouteTableAssociationCreated").
            value(
                    String.format(
                            "SubnetId: %s, RouteTableId: %s",
                            subnetRouteTableAssociation.getSubnetId(),
                            subnetRouteTableAssociation.getRouteTableId()
                    )
            ).
            build();
    return publicRouteTable;
}
```

Amazon AWS Cloud Development Kit (CDK)

Challenge

Create the following using your CDK stack:

- NatGateway
- Private RouteTable with Routes and the SubnetRouteTableAssociation

😎 GOOD LUCK 😎

Amazon AWS Cloud Development Kit (CDK)

Destroy

- To destroy the stack use the following command:

  ```
  cdk destroy
  ```