

Track 4 | Session 5

架構即代碼 – AWS CDK 與 CDK8S 聯手打造下一代的 K8S 應用

Pahud Hsieh

Developer Advocate
Amazon Web Services

想像一下

We Need to Build A
Huge Building
Like This



Our Teams

- Owner (You)
 - You can control everything
- Designer/Architect (設計師/建築師)
 - Structural, internal and external design
 - Compliance
 - Security
- Construction(施工團隊)
 - working with the building blocks by design

How the construction team works?

Construction Team

Working with Mouse Clicks in the AWS Console

aws

Services

Resource Groups

🔔

pahud @ aws-pahud

Tokyo

Support

New EC2 Experience

Tell us what you think

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

Lifecycle Manager

NETWORK & SECURITY

Security Groups

Elastic IPs

Placement Groups

EC2

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Tokyo) Region:

Running instances	0	Elastic IPs	2	Dedicated Hosts	0
Snapshots	0	Volumes	2	Load balancers	11
Key pairs	3	Security groups	97	Placement groups	0

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance

Note: Your instances will launch in the Asia Pacific (Tokyo) Region

Scheduled events

Asia Pacific (Tokyo)

No scheduled events

Migrate a machine

Use CloudEndure Migration to simplify, expedite, and automate large-scale

Service health

Region

Asia Pacific (Tokyo)

Status

🟢 This service is operating normally

Zone status

Zone	Status
ap-northeast-1a (apne1-az4)	🟢 Zone is operating normally
ap-northeast-1c (apne1-az1)	🟢 Zone is operating normally
ap-northeast-1d (apne1-az2)	🟢 Zone is operating normally

Enable additional Zones

Account attributes

Supported platforms

VPC

Default VPC

vpc-497a492d

Settings

EBS encryption

Zones

Console experiments

Explore AWS

Save up to 90% on EC2 with Spot Instances

Optimize price-performance by combining EC2 purchase options in a single EC2 ASG. Learn more

Easily launch third-party AMI products

AWS Marketplace has thousands of third-party AMI products that you can find, buy, and deploy with 1-click using the Amazon EC2 console. Learn more

Save 10% with AMD EPYC-Powered Instances

Lower cost on compute and memory with AMD EPYC processors. Learn more

Construction Workers

AWS CLI or Shell Scripts

```
#!/bin/bash
```

```
# create vpc
```

```
aws ec2 create-vpc \  
--ipv6-cidr-block-network-border-group us-west-2-lax-1 \  
--cidr-block 10.0.0.0/16
```

```
# create subnets
```

```
# create internet gateway and nat gateways
```

```
# create routing table and routes
```

```
# create ec2 instances
```

```
aws ec2 run-instances --image-id ami-xxxxxxx --count 1 \  
--instance-type t2.micro \  
--key-name MyKeyPair --security-group-ids sg-903004f8 \  
--subnet-id subnet-6e7f829e
```

Designer and Architect?

Designer and Architect

Using Templates

```
"Resources" : {
  "EC2Instance" : {
    "Type" : "AWS::EC2::Instance",
    "Properties" : {
      "InstanceType" : { "Ref" : "InstanceType" },
      "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
      "KeyName" : { "Ref" : "KeyName" },
      "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
                                     { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ] } ] }
    }
  },

  "InstanceSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
      "GroupDescription" : "Enable SSH access via port 22",
      "SecurityGroupIngress" : [ {
        "IpProtocol" : "tcp",
        "FromPort" : "22",
        "ToPort" : "22",
        "CidrIp" : { "Ref" : "SSHLocation" }
      } ]
    }
  },
},
```

Designer and Architect

Using Templates

It could be...

Thousands of lines in JSON or YAML

What Do You Want?

Give me a **VPC** out-of-the-box

```
new ec2.Vpc();
```

Give me a **EC2** in this VPC

```
new ec2.Instance(stack, 'instance', {  
    vpc,  
});
```


Give me a **S3 bucket** out-of-the-box

```
new s3.Bucket();
```

Grant the EC2 instance role to read/write the bucket

```
const vpc = new ec2.Vpc();  
const instance = new ec2.Instance(this, 'Instance', { vpc, ...});  
const bucket = new s3.Bucket();  
bucket.grantReadWrite(instance.role);
```

And it compiles into this

```
1 Resources:
2   Bucket83908E77:
3     Type: AWS::S3::Bucket
4     UpdateReplacePolicy: Retain
5     DeletionPolicy: Retain
6     Metadata:
7       aws:cdk:path: XxxStack/Bucket/Resource
8   Vpc8378EB38:
9     Type: AWS::EC2::VPC
10    Properties:
11      CidrBlock: 10.0.0.0/16
12      EnableDnsHostnames: true
13      EnableDnsSupport: true
14      InstanceTenancy: default
15      Tags:
16        - Key: Name
17          Value: XxxStack/Vpc
18      Metadata:
19        aws:cdk:path: XxxStack/Vpc/Resource
20   VpcPublicSubnet1Subnet5C2D37C4:
21     Type: AWS::EC2::Subnet
22     Properties:
23       CidrBlock: 10.0.0.0/18
24       VpcId:
25         Ref: Vpc8378EB38
26       AvailabilityZone:
27         Fn::Select:
28           - 0
29           - Fn::GetAZs: ""
30       MapPublicIpOnLaunch: true
31       Tags:
32         - Key: aws-cdk:subnet-name
33           Value: Public
34         - Key: aws-cdk:subnet-type
35           Value: Public
36         - Key: Name
37           Value: XxxStack/Vpc/PublicSubnet1
38       Metadata:
39         aws:cdk:path: XxxStack/Vpc/PublicSubnet1/Subnet
40   VpcPublicSubnet1RouteTable6C95E38E:
41     Type: AWS::EC2::RouteTable
42     Properties:
43       VpcId:
44         Ref: Vpc8378EB38
45       Tags:
46         - Key: Name
47           Value: XxxStack/Vpc/PublicSubnet1
48       Metadata:
49         aws:cdk:path: XxxStack/Vpc/PublicSubnet1/RouteTable
50   VpcPublicSubnet1RouteTableAssociation97140677:
51     Type: AWS::EC2::SubnetRouteTableAssociation
52     Properties:
```

And takes care of this

```
InstanceInstanceRoleDefaultPolicy4ACE9290:
  Type: AWS::IAM::Policy
  Properties:
    PolicyDocument:
      Statement:
        - Action:
            - s3:GetObject*
            - s3:GetBucket*
            - s3:List*
            - s3:DeleteObject*
            - s3:PutObject*
            - s3:Abort*
          Effect: Allow
          Resource:
            - Fn::GetAtt:
                - Bucket83908E77
                - Arn
            - Fn::Join:
                - ""
                - - Fn::GetAtt:
                    - Bucket83908E77
                    - Arn
                - /*
          Version: "2012-10-17"
    PolicyName: InstanceInstanceRoleDefaultPolicy4ACE9290
    Roles:
      - Ref: InstanceInstanceRoleE9785DE5
```

Minimal required IAM Policy

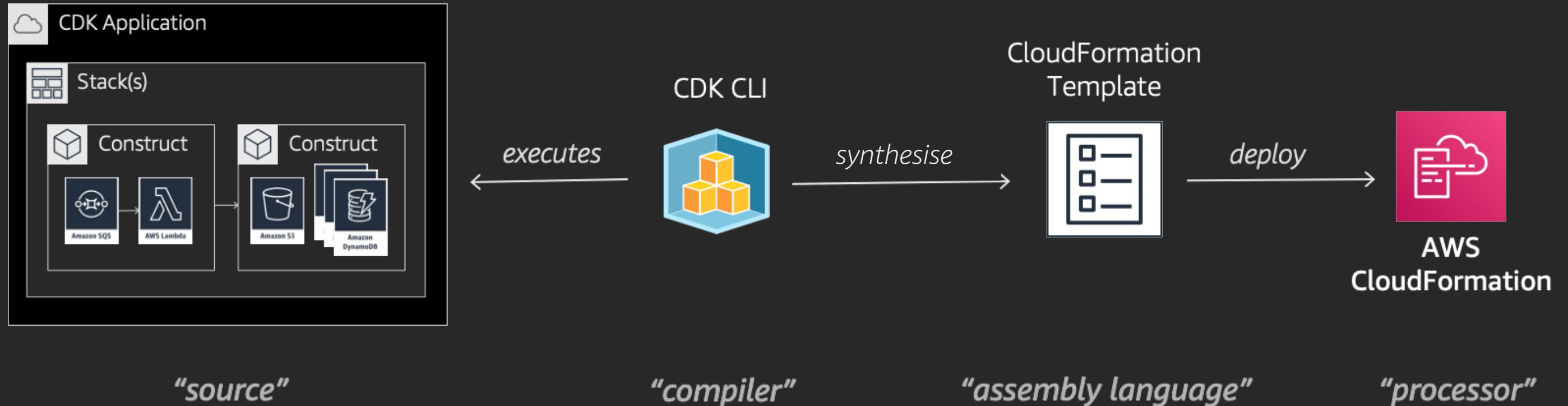
```
InstanceInstanceProfileAB5AEF02:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Roles:
      - Ref: InstanceInstanceRoleE9785DE5
  Metadata:
    aws:cdk:path: XxxStack/Instance/InstanceProfile
```

IAM Instance Profile

```
InstanceC1063A87:
  Type: AWS::EC2::Instance
  Properties:
    AvailabilityZone:
      Fn::Select:
        - 0
        - Fn::GetAZs: ""
    IamInstanceProfile:
      Ref: InstanceInstanceProfileAB5AEF02
    ImageId:
      Ref: SsmParameterValueawsserviceamazonlinuxlatestamznihvmx8664gp2C96584B6F00A464EAD1953AFF4B05118Parameter
    InstanceType: t3.large
    SecurityGroupIds:
      - Fn::GetAtt:
          - InstanceInstanceSecurityGroupF0E2D5BE
```

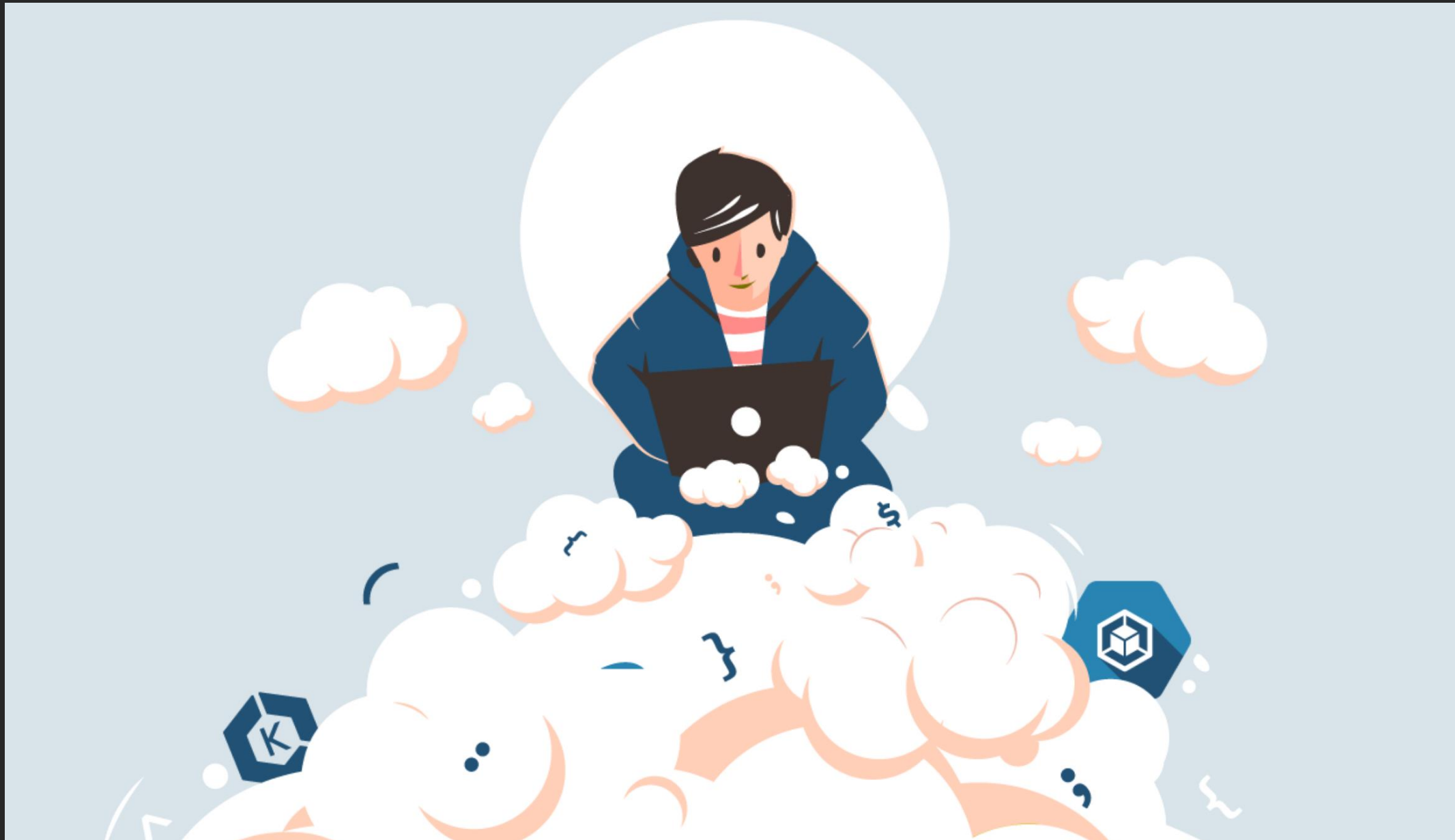
What is the AWS Cloud Development Kit (AWS CDK)

The big picture - from AWS CDK app to CloudFormation to provisioned infrastructure



CDK for Kubernetes (CDK8S)

Define Kubernetes apps and components using familiar languages



AWS CDK + CDK8S – What is the Experience

Everything Under the Kubernetes – **AWS CDK**

All cloud infrastructure

- VPC Networking, Amazon EKS Cluster, Managed Nodegroup, Fargate Profile, etc.

Everything Above the Kubernetes – **CDK8S**

Kubernetes Resources

- deployment, service, daemonset, ingress, namespaces, pod, roles, rolebindings, jobs, etc.

Everything You Need

- Your Favorite IDE with Your Favorite Language

Demo

Thank you!

Pahud Hsieh
Developer Advocate

 @pahudnet