白皮书：

基于 AWS CDK 的跨云迁移实践

—— 从阿里云到 AWS 新加坡区域

发布方：成都易定云科技有限公司
日　期：2026年3月
版　本：1.0

摘要

随着企业全球化业务拓展，云平台的选择与迁移成为技术决策的关键。成都易定云科技有限公司（以下简称“易定云”）协助一家长期服务于广告 SAAS 行业的客户，成功将亚太区业务从阿里云新加坡区域平滑迁移至 AWS 新加坡（ap-southeast-1）及东京（ap-northeast-1）灾备区域。

本次迁移采用 AWS CDK (Java) 作为基础设施即代码（IaC）工具，构建了 7 个模块化 Stack，实现了 145+ CloudFormation 资源的全自动化部署。通过遵循 AWS Well-Architected Framework，在卓越运营、安全性、可靠性、性能效率、成本优化五大支柱上全面升级，充分体现了易定云在云原生架构设计与自动化运维领域的深厚积累。

一、背景与挑战

在客户原业务部署于阿里云新加坡区域，随着业务增长，面临以下挑战：

基础设施管理复杂

部分资源依赖手动创建，缺乏统一的代码化定义，导致环境一致性差、变更风险高。

安全合规要求提升

需满足 HIPAA 等严格审计要求，原有环境在加密、日志审计、访问控制方面存在不足。

灾备能力不足

缺乏跨区域容灾机制，业务连续性保障薄弱。

运维效率瓶颈

监控、告警、日志分析分散，故障定位困难，运维成本高。

二、易定云解决方案

易定云基于 AWS CDK（v2.232.1）设计并实施了一套端到端的现代化基础设施，实现从阿里云至 AWS 新加坡区域的主站点部署，并同步构建东京区域灾备环境。

2.1 CDK 模块化架构

项目采用 Java 编写 CDK 应用，通过 7 个独立 Stack 实现关注点分离：
Stack采用模块化设计，结构如下：

```
AwsInfrastructureCdkApp (入口)
├── NetworkStack          — VPC、子网、NAT、IGW、VPC Endpoints、Flow Logs
├── SecurityStack         — KMS、IAM Roles、WAF Web ACL、Security Groups
├── DataStack             — Aurora MySQL、ElastiCache Redis、S3、ECR
├── ComputeStack          — EKS 集群、Bastion Host、Lambda 自动化函数
├── EdgeStack             — CloudFront、Global Accelerator、Route 53
├── ObservabilityStack    — CloudWatch、CloudTrail、SNS 告警
└── DrStack               — DR 区域 VPC、S3 备份、ECR 镜像仓库
```

Stack依赖关系如下：

```
NetworkStack
    └── SecurityStack
        ├── DataStack
        │       └── ObservabilityStack
        ├── ComputeStack
        └── EdgeStack
DrStack (独立部署至东京区域)
```

易定云见解：
模块化 Stack 设计不仅便于团队并行开发，还能在 CI/CD 流水线中实现细粒度部署控制，有效降低变更影响范围。

2.2核心技术亮点

我们利用AWS CDK实现了“一次定义，多地部署”，具体步骤如下：

高可用网络设计

新加坡主区域采用双可用区部署，VPC CIDR 规划合理，公网子网、应用私有子网、数据库私有子网严格隔离。东京灾备区域通过 DrStack 自动创建独立 VPC 及关键存储资源，并开启 VPC Flow Logs 实现网络流量可审计。

全栈加密与安全加固

KMS 主密钥启用自动轮转，所有存储服务（S3、Aurora、ECR）默认加密。WAF Web ACL 关联 CloudFront，启用 3 组 AWS 托管规则集抵御常见 Web 攻击。IAM 角色遵循最小权限原则，通过 CDK 代码精确定义权限边界。Security Groups 分层管理，仅开放必要端口。

数据层与容器平台

数据库采用 Aurora MySQL 集群，支持加密、自动备份、跨可用区部署；缓存使用 ElastiCache Redis 集群，用于会话存储与数据缓存。容器平台基于 Amazon EKS (Kubernetes 1.33)，结合托管节点组与 Fargate 满足不同负载需求。堡垒机仅通过 SSM Session Manager 访问，不对外开放 22 端口。

边缘优化与加速

CloudFront 实现全球内容分发加速并与 WAF 联动；Global Accelerator 优化 TCP/UDP 访问链路，降低延迟；Route 53 提供智能解析与主备切换能力。

可观测性与运维自动化

CloudWatch Dashboard 统一展示核心资源指标，CloudWatch Alarms 实现异常自动告警并通过 SNS 推送。CloudTrail 开启多区域审计，日志自动归档至 S3 并配置生命周期策略，Container Insights 实现 EKS 容器性能监控。

跨区域灾备自动化

DrStack 在东京区域自动部署 VPC、S3 备份桶、ECR 镜像仓库；通过 S3 跨区域复制（CRR）实现数据自动备份，ECR 跨区域复制保证镜像在灾备区域可用。

三、自动化部署流程

易定云将整个基础设施的部署过程代码化，实现“一键部署”与“分步可控”的双重能力。

3.1环境初始化

首次部署需在两个区域执行 CDK 引导（Bootstrap）：

```
cd aws-infrastructure-cdk
cdk bootstrap aws://<ACCOUNT_ID>/ap-southeast-1  # 新加坡
cdk bootstrap aws://<ACCOUNT_ID>/ap-northeast-1  # 东京
```

```
Trusted accounts for deployment: (none)
Trusted accounts for lookup: (none)
Using default execution policy of 'arn:aws:iam::aws:policy/AdministratorAccess'. Pass '--cloudformation-execution-policies
' to customize.
CDKToolkit: creating CloudFormation changeset...
[                                  ·················] (8/12)

5:48:17 AM | CREATE_IN_PROGRESS   | AWS::CloudFormation::Stack | CDKToolkit
5:48:41 AM | CREATE_IN_PROGRESS   | AWS::IAM::Policy           | ImagePublishingRoleDefaultPolicy
5:48:41 AM | CREATE_IN_PROGRESS   | AWS::IAM::Policy           | FilePublishingRoleDefaultPolicy
5:48:42 AM | CREATE_IN_PROGRESS   | AWS::IAM::Role             | DeploymentActionRole
```

3.2逐 Stack 部署（推荐）

通过依次部署各 Stack，可清晰观测资源创建过程，便于故障定位：

```
cdk deploy NetworkStack
cdk deploy SecurityStack
cdk deploy DataStack
cdk deploy ComputeStack
cdk deploy EdgeStack
cdk deploy ObservabilityStack
cdk deploy DrStack
```

```
NetworkStack: deploying... [1/1]
NetworkStack: creating CloudFormation changeset...

 ✅  NetworkStack

✨  Deployment time: 157.47s

Outputs:
NetworkStack.ExportsOutputFnGetAttPrimaryVpc0877506ECidrBlockDB6A8A83 = 10.0.0.0/16
NetworkStack.ExportsOutputRefPrimaryVpc0877506E0840D8DD = vpc-07a0[redacted]c5fc
NetworkStack.ExportsOutputRefPrimaryVpcIsolatedSubnet1SubnetEFBCAEA21B7E96F5 = subnet-00f3e965a0ee77535
NetworkStack.ExportsOutputRefPrimaryVpcIsolatedSubnet2SubnetA9B8F422B2E321A1 = subnet-0523ff516558fca06
NetworkStack.ExportsOutputRefPrimaryVpcPrivateSubnet1Subnet53BC8B69899BE70B = subnet-0fa50d826b5ad98de
NetworkStack.ExportsOutputRefPrimaryVpcPrivateSubnet2Subnet2573FA4E4C060539 = subnet-07b473347f9bc0534
NetworkStack.ExportsOutputRefPrimaryVpcPublicSubnet1Subnet7A950DC04FFED2DD = subnet-0a663cf918c662688
Stack ARN:
arn:aws:cloudformation:ap-southeast-1[redacted]tack/NetworkStack/fa32d0f0-28[redacted]088491451

✨  Total time: 171.37s
```

```
SecurityStack: deploying... [2/2]
SecurityStack: creating CloudFormation changeset...

 ✅  SecurityStack

✨  Deployment time: 50.96s

Outputs:
SecurityStack.ExportsOutputFnGetAttMasterKey0175C9E2Arn2B8655C7 = arn:aws:kms:ap-southeast-1:[redacted]ey/c734219e-268[redacted]a651f40f68d
Stack ARN:
arn:aws:cloudformation:ap-southeast-[redacted]tack/SecurityStack/83e9916[redacted]02d7e50ef809

✨  Total time: 64.76s
```

```
EdgeStack: deploying... [1/1]
EdgeStack: creating CloudFormation changeset...

 ☑  EdgeStack

✨  Deployment time: 279.36s

Stack ARN:
arn:aws:cloudformation:ap-southeast-1:[redacted]stack/EdgeStack/0d0205[redacted]7-0a68dfdbb2c7

✨  Total time: 293.11s
```

```
DataStack: deploying... [3/3]
DataStack: creating CloudFormation changeset...

 ☑  DataStack

✨  Deployment time: 587.55s

Outputs:
DataStack.ExportsOutputRefLogBucketCC3B17E818DCEC53 = migration-logs-[redacted]
Stack ARN:
arn:aws:cloudformation:ap-southeast-1:[redacted]tack/DataStack/25faa4f0-[redacted]02a4ca8fddd3

✨  Total time: 601.63s
```

```
ObservabilityStack
ObservabilityStack: deploying... [4/4]
ObservabilityStack: creating CloudFormation changeset...

 ☑  ObservabilityStack

✨  Deployment time: 24.39s

Stack ARN:
arn:aws:cloudformation:ap-southeast-1:[redacted]stack/ObservabilityStack/b50465[redacted]efdca483b

✨  Total time: 38.22s
```

```
DrStack: deploying... [1/1]
DrStack: creating CloudFormation changeset...

 ☑  DrStack

✨  Deployment time: 158.17s

Stack ARN:
arn:aws:cloudformation:ap-northeast-1:[redacted]tack/DrStack/fa160f70-28fe-11f1-b380-0e590332d33d

✨  Total time: 172.34s
```

易定云见解：

通过 CDK context 参数（如 –c environment=prod），同一套代码可支持开发、测试、生产环境的差异化配置，实现环境一致性。

四、核心实践与价值体现

4.1基础设施即代码（IaC）的深度应用

100% 代码化

所有 AWS 资源均通过 Java CDK 定义，无手动控制台操作，杜绝配置漂移。

版本控制

CDK 代码纳入 Git，支持变更审计、回滚与协作。

可重复部署

cdk deploy --all 一条命令即可完成全部资源部署，大幅缩短环境交付周期。

4.2安全与合规内建

加密自动化：KMS 主密钥自动轮换，所有存储服务强制加密，满足 HIPAA 数据保护要求。WAF 防护：CloudFront 前置 WAF，托管规则集自动拦截 SQL 注入、XSS 等攻击。审计追踪：CloudTrail 多区域开启，日志保留周期自定义，满足合规审计需求。

4.3高可用与灾备

主区域双 AZ 部署，关键服务跨可用区冗余，单 AZ 故障不影响业务。跨区域灾备通过 DrStack 在东京自动构建基础设施，S3/ECR 数据自动复制，RTO/RPO 可控。

4.4运维效率提升

统一监控：CloudWatch Dashboard 聚合关键指标，减少控制台登录操作。自动告警：核心指标超阈值自动触发 SNS 通知，支持钉钉 / 邮件集成。日志生命周期：S3 日志桶 90 天归档至 Glacier，365 天自动过期，降低存储成本。

五、成果与客户收益

基础设施交付效率提升 80%

从数周手工配置缩短至数小时 CDK 部署。

安全合规全面达标

通过加密、审计、WAF 等机制，满足 HIPAA 及企业内部安全规范。

灾备能力从无到有

东京区域灾备环境自动化构建，业务连续性得到保障。

运维成本显著降低

自动化监控、告警与日志管理减少人工干预，故障定位时间缩短 60%。

六、总结

本次从阿里云向 AWS 的跨云迁移，充分验证了成都易定云科技有限公司在云原生架构设计、基础设施即代码、自动化运维领域的领先能力。通过 AWS CDK 的模块化实践，我们为客户构建了一套安全合规、高可用、可弹性扩展、可观测的现代化基础设施。

未来

易定云将继续深耕 AWS 技术栈
结合 CDK、EKS、Serverless 等前沿服务
助力更多企业实现云上卓越运营。

FUTURE