白皮书：

基于 AWS CDK 的 EKS 自动化部署实践

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

随着企业云原生转型的深入，Amazon EKS 已成为 Kubernetes 托管服务的首选。然而，EKS 集群的创建与管理涉及 VPC 网络规划、IAM 权限配置、OIDC 集成、节点组弹性伸缩等多个复杂环节。成都易定云科技有限公司（以下简称“易定云”）基于 AWS CDK（Python）构建了一套模块化的 EKS 自动化部署方案，实现了集群与节点组的分离定义、灵活配置与一键部署。

本白皮书深入解析易定云在 EKS 基础设施即代码领域的实践经验，涵盖 VPC 复用策略、ARM/x86 混合架构支持、IRSA 安全集成、节点组动态扩展等核心技术，为企业快速落地云原生平台提供可复用的参考方案。

一、背景与挑战

企业规模化落地 Amazon EKS 集群时，普遍面临以下行业痛点：

部署流程复杂

手动创建 EKS 集群需同步配置 VPC、子网、IAM 角色、安全组等多项资源，配置链路长、人工操作多，极易引发配置错误。

环境一致性差

开发、测试、生产环境依靠手工差异化配置，长期运行易产生环境漂移问题，引发上线故障。

节点组管理不灵活

不同业务场景适配不同实例规格、CPU 架构（ARM/x86）、资源扩容策略，缺乏统一标准化配置入口

扩展组件集成繁琐

AWS Load Balancer Controller、EBS CSI Driver 等主流插件依赖 OIDC 与 IRSA 关联配置，手工配置门槛高、合规风险大。

易定云见解：

EKS 基础设施管理应当严格遵循集群与节点组分离原则：集群底层架构生命周期稳定，无需频繁变更；节点组需跟随业务流量、容器负载动态弹性调整。两者解耦后可独立管控生命周期，极大提升云原生运维整体灵活性与稳定性。

二、易定云解决方案

易定云基于 AWS CDK（Python）自研模块化 EKS 自动化部署解决方案，标准化封装底层复杂资源，支撑企业轻量化一键交付。

2.1 核心代码架构

```
CDK项目目录/
├── app.py                    # CDK应用统一入口
├── eks_stack.py              # EKS集群核心Stack（VPC、集群、默认节点组）
├── nodegroup_stack.py        # 独立节点组管理Stack（存量集群扩容新增节点）
└── .venv/                    # Python独立虚拟运行环境
```

核心设计思想

模块化 Stack 拆分EksStack

负责全新 VPC 创建 / 存量 VPC 复用、EKS 管控集群部署、基础节点组初始化、负载均衡组件 IAM 权限预置；NodegroupStack：独立解耦，专为已有在线集群灵活新增异构节点组，适配多业务算力需求。

全局参数化配置

依托 CDK Context 全局传参，同一套底层代码无缝适配开发、测试、生产多套部署场景。

原生安全自动集成

自动化完成 OIDC 身份提供商创建、IRSA 权限关联，为负载均衡控制器绑定服务账号与 IAM 权限体系。

2.2 核心技术能力

2.2.1 灵活双模 VPC 管理

EKS 集群资源栈支持新建 VPC、复用存量 VPC两种标准化模式：

新建 VPC

未传入vpc_id参数时，自动构建双可用区专属 VPC，配置单 NAT 网关，自动为公私网子网标注 EKS 专属资源标签；

复用存量 VPC

传入现有vpc_id，直接对接企业已有网络架构，杜绝资源重复创建、节约成本。

```
# VPC自动创建/存量复用核心逻辑代码
if vpc_id:
    vpc = ec2.Vpc.from_lookup(self, "ExistingVpc", vpc_id=vpc_id)
else:
    vpc = ec2.Vpc(self, "EksVpc", vpc_name=vpc_name, max_azs=2, nat_gateways=1)
```

2.2.2 ARM/X86 混合异构节点组支持

独立节点组资源栈一键切换两大主流 CPU 架构：

ARM 架构

匹配镜像 AL2023_ARM_64_STANDARD，适配 Graviton 系列实例；

X86 架构

匹配镜像 AL2023_X86_64_STANDARD，兼容传统通用算力实例；通过instance_type参数自定义配置机型（m7g.large、m5.large 等），灵活适配业务。

易定云见解：
AWS Graviton ARM 架构实例针对容器化场景深度优化，性价比远超传统 x86 架构，落地后可降低 **20%–40%** 计算资源成本；同步保留 x86 架构兼容入口，适配老旧应用、特殊中间件等异构场景。

节点组支持两类子网一键切换部署：

2.2.3 子网智能分类部署

节点组支持两类子网一键切换部署：

私有子网（默认推荐）

承载核心业务 Pod，依托 NAT 网关安全访问外网，缩减公网暴露攻击面；

公网子网

适配堡垒机、特殊外网直通服务等小众业务场景；通过subnet_type参数（private/public）快速配置，无需修改底层代码。

2.2.4 AWS Load Balancer Controller 自动化 IRSA 集成

集群资源栈预置标准化配置，全自动完成负载均衡控制器权限编排：

1.自动创建命名空间kube-system下专属服务账号 aws-load-balancer-controller；
2.绑定 ELB 官方托管权限策略，满足流量转发、资源调度需求；
3.输出 IAM 角色 ARN，无缝对接后续 Helm 应用部署流程。
python

```
# 负载均衡控制器IRSA绑定核心代码
sa = cluster.add_service_account("AwsLbControllerSA",
    name="aws-load-balancer-controller",
    namespace="kube-system",
)
sa.role.add_managed_policy(
    iam.ManagedPolicy.from_aws_managed_policy_name("ElasticLoadBalancingFullAccess")
)
```

2.2.5 节点组精细化动态弹性配置

支持运维可视化自定义核心运行参数，满足全场景算力调度：

参数	说明	典型配置示例
min_size	节点池最小保底实例数	2
max_size	节点池最大扩容上限实例数	10
desired_size	初始化期望运行节点数量	2
instance_type	EC2 算力实例规格型号	m7g.large
disk_size	节点系统盘存储空间 (GB)	50
ami_type	操作系统 CPU 架构类型	ARM / X86
subnet_type	节点部署子网网络类型	private / public

三、自动化标准部署流程

3.1 前置环境准备命令集

```
# 进入CDK工程根目录
cd CDK

# 激活Python隔离虚拟环境
source .venv/bin/activate

# 安装项目核心依赖包
pip install aws-cdk-lib constructs aws-cdk.lambda-layer-kubectl-v31

# 全局升级CDK客户端至最新稳定版本
npm install -g aws-cdk@latest

# 配置AWS云服务商访问密钥（首次部署必操作）
aws configure

# 账户+区域初始化Bootstrap环境（每个AWS账号单区域仅执行一次）
cdk bootstrap
```

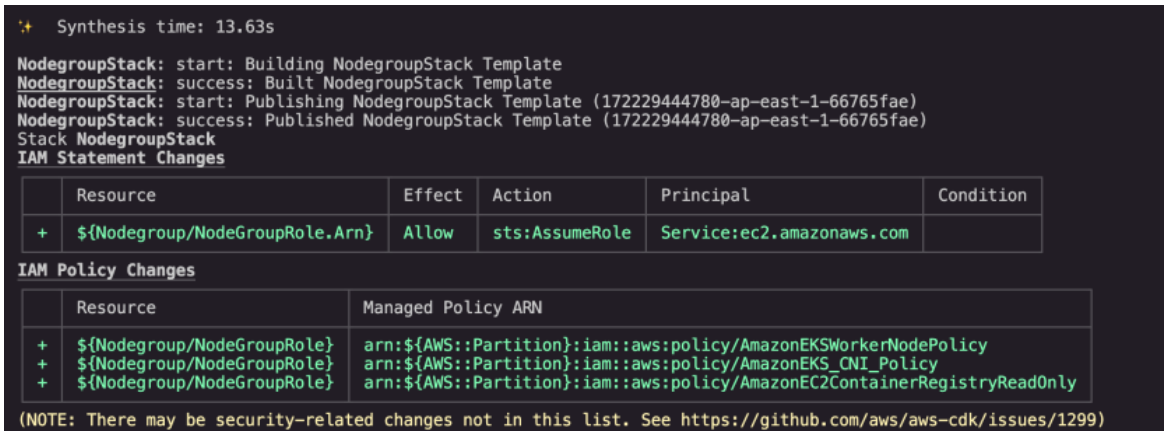
3.2 四大主流落地部署场景

场景一：全新构建专属 VPC+EKS 管控集群

```
cdk deploy -c cluster_name=my-cluster -c vpc_name=my-vpc
```

场景二：复用企业现有存量 VPC 部署 EKS 集群

```
cdk deploy -c cluster_name=my-cluster -c vpc_id=vpc-xxxx
```



场景三：存量 EKS 集群新增 ARM 架构弹性节点组

```
cdk deploy NodegroupStack \
  -c cluster_name=my-cluster \
  -c vpc_id=vpc-xxxx \
  -c nodegroup_name=private-nodes \
  -c instance_type=m7g.large \
  -c min_size=2 \
  -c max_size=5
```

场景四：存量 EKS 集群新增 X86 架构兼容节点组

```
cdk deploy NodegroupStack \
  -c cluster_name=my-cluster \
  -c vpc_id=vpc-xxxx \
  -c nodegroup_name=x86-nodes \
  -c instance_type=m5.large \
  -c ami_type=X86
```

3.3 集群部署标准化验证

```
# 本地配置集群访问凭证，关联管理员角色
aws eks update-kubeconfig --name <集群名> --role-arn <MastersRole ARN>

# 校验所有工作节点运行状态
kubectl get nodes

# 校验负载均衡控制器专属服务账号配置
kubectl get sa -n kube-system aws-load-balancer-controller
```

四、易定云核心落地实践方法论

4.1 集群与节点组解耦独立管理

传统架构将集群、节点组固化在同一资源栈，节点变更极易引发集群底层配置扰动。方案拆分双独立 Stack 后实现：

变更隔离

节点池扩容、缩容、删除操作，零感知不影响核心管控集群稳定性；

权限拆分

集群管理员、节点运维管理员配置独立 IAM 权限，满足企业分级管控；

生命周期独立

节点组按需弹性增减，无需重构 EKS 基础集群。

4.2 全域参数化统一配置体系

基于 CDK Context 上下文全局读取自定义业务参数，实现一套代码、多环境复用：

```
# 全局默认参数读取逻辑
cluster_name = self.node.try_get_context("cluster_name") or "my-eks-cluster"
instance_type = self.node.try_get_context("instance_type") or "m7g.large"
```

不同环境通过独立参数文件差异化调度，杜绝多代码分支管理、减少运维混乱。

4.3 底层安全自动化原生内嵌

1.IAM 最小权限原则：仅预置集群创建、节点运维必需权限，杜绝超范围授权风险；
2.OIDC 免手工配置：CDK 底层自动化创建 IAM OIDC 身份信任提供商；
3.IRSA 全自动关联：服务账号与云资源 IAM 角色一键绑定，消除人工配置疏漏。

4.4 企业级 EKS 架构专项优化建议

优化分类	落地推荐实践	业务价值收益
成本优化	优先选用 Graviton ARM 系列机型（m7g/c7g）	综合算力成本降低 20%–40%
弹性调度	接入 Karpenter 智能伸缩，替代静态固定节点组	服务器资源利用率提升 30%+
高可用保障	节点组跨多可用区部署，最小节点数≥2	单可用区故障自动容错，业务无中断
存储性能	统一使用 gp3 极速型 EBS 云盘，自定义配比 IOPS	存储成本下降 20%，性能按需可调
安全加固	业务节点全域部署私有子网，NAT 网关统一出站	大幅缩减公网攻击暴露面，满足合规

五、落地成果与客户商业价值

部署效率提升 80%

传统数小时人工配置流程，压缩至分钟级代码一键自动化交付；

环境强一致性保障

研测生全环境同源代码部署，彻底根治配置漂移、上线突发故障；

常态化运维成本压降

节点组参数化标准化管理，弹性策略可视化调度，减少人工值守干预；

安全合规原生内置

IAM 最小权限、OIDC/IRSA 自动编排、私有子网隔离等安全策略默认启用；

混合算力全域兼容

ARM 节能算力与 X86 通用算力混合部署，兼顾成本优化与老旧应用兼容性。

六、总结与未来规划

成都易定云科技有限公司依托 AWS CDK（Python）打造轻量化、模块化 EKS 全自动化部署体系，通过资源栈解耦设计、参数化全域调度、异构算力兼容三大核心能力，根本性解决企业 EKS 云原生落地复杂、配置杂乱、合规薄弱等痛点。
该解决方案已规模化落地各行业生产环境，助力客户快速搭建高可用、高安全、低成本的企业级云原生容器平台。

未来

展望未来，易定云将
持续深耕GitOps
持续交付、Karpenter 极致弹性伸缩、多云多集群统一管控核心技术赛道
持续迭代产品能力
赋能各行业企业实现云原生基础设施精益化、智能化卓越运营。

FUTURE