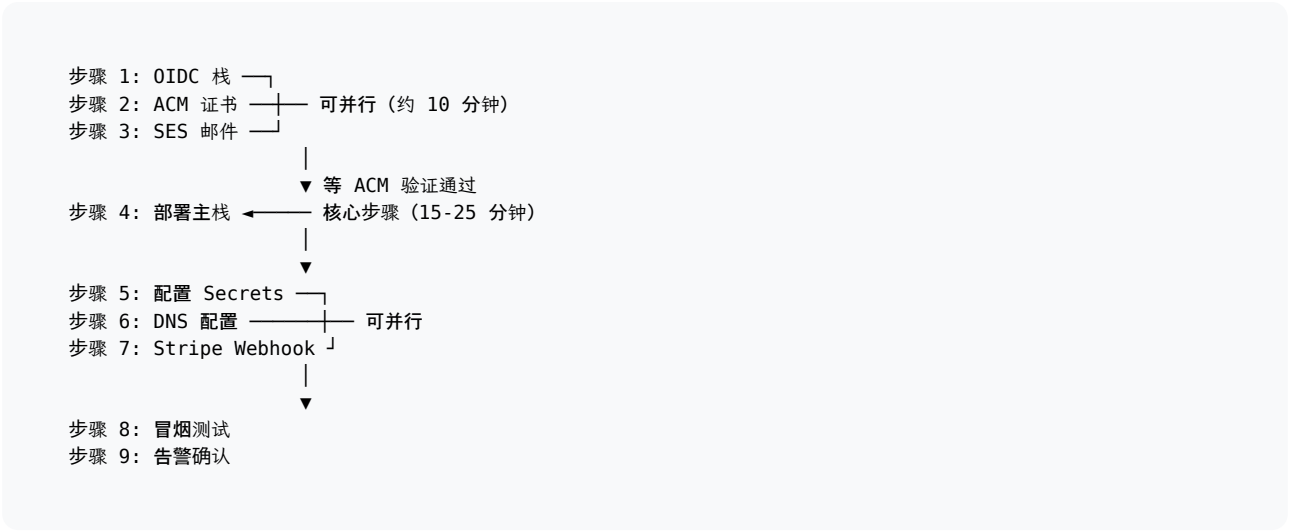
Crab SaaS

内测上线操作手册

AWS eu-south-2 · Cloudflare DNS · GitHub Actions CI/CD
版本: 2026-02-18 · 预计执行时间: 1–2 小时

执行概览

本手册包含 9 个步骤，将 crab-auth (Lambda) 和 crab-cloud (ECS Fargate) 部署到 AWS 生产环境。



准备清单

项目	说明	状态
AWS CLI	已安装且配置 eu-south-2 区域	☐
AWS 账号	有 IAM 管理权限	☐
域名	Cloudflare DNS 托管	☐
Stripe	已有账号，有 API Key	☐
GitHub	repo 有 Settings 权限	☐
代码	已 push 最新 commit (含 OIDC 模板)	☐

第 1 步　部署 GitHub OIDC 栈

让 GitHub Actions 通过 OIDC 免密认证 AWS，不需要存储 Access Key。

1.1 创建 Lambda S3 Bucket

```
aws s3 mb s3://crab-deploy-artifacts --region eu-south-2
```

✓ 输出: `make_bucket: crab-deploy-artifacts`

1.2 部署 OIDC CloudFormation 栈

```
aws cloudformation deploy \
  --template-file deploy/github-oidc.yml \
  --stack-name crab-github-oidc \
  --capabilities CAPABILITY_IAM \
  --region eu-south-2 \
  --parameter-overrides \
    GitHubOrg=你的GitHub用户名 \
    GitHubRepo=你的repo名
```

等待完成（约 2 分钟）。

1.3 获取 Role ARN

```
aws cloudformation describe-stacks \
  --stack-name crab-github-oidc \
  --query 'Stacks[0].Outputs[?OutputKey==`DeployRoleArn`].OutputValue' \
  --output text \
  --region eu-south-2
```

✓ 记录输出的 ARN: `arn:aws:iam::XXXX:role/crab-github-deploy`

1.4 设置 GitHub Secrets

打开 GitHub repo → **Settings** → **Secrets and variables** → **Actions**，添加：

Secret Name	值
`AWS_DEPLOY_ROLE_ARN`	上一步获取的 Role ARN
`LAMBDA_S3_BUCKET`	`crab-deploy-artifacts`

TIP 其他 Secrets（TAURI 签名、UPDATE_S3 等）内测阶段暂不需要。

第 2 步　ACM 证书 (HTTPS)

注意 此步骤可与步骤 1、3 **并行**执行。

2.1　申请通配符证书

```
aws acm request-certificate \
  --domain-name "你的域名" \
  --subject-alternative-names "*.你的域名" \
  --validation-method DNS \
  --region eu-south-2
```

✓ 记录输出的 `CertificateArn`，步骤 4 需要。

2.2　获取 DNS 验证记录

```
aws acm describe-certificate \
  --certificate-arn 上一步的ARN \
  --query 'Certificate.DomainValidationOptions[].ResourceRecord' \
  --output table \
  --region eu-south-2
```

输出一条 CNAME 记录 (Name + Value)。

2.3　在 Cloudflare 添加验证 CNAME

Cloudflare Dashboard → 你的域名 → **DNS** → **Add record**:

Type	CNAME
Name	ACM 给的 Name（去掉你的域名后缀部分）
Target	ACM 给的 Value
Proxy	❌ DNS only（灰色云）

2.4　等待验证通过

```
aws acm wait certificate-validated \
  --certificate-arn 你的ARN \
  --region eu-south-2
```

命令无输出即成功。通常需要 5–10 分钟。

2.5　验证

```
aws acm describe-certificate \
  --certificate-arn 你的ARN \
```

```
  --region eu-south-2 \
  --query 'Certificate.Status'
# 输出: "ISSUED"
```

注意 证书必须在 `eu-south-2` 区域申请，与部署区域一致。

第 3 步 SES 邮件服务

用于发送注册验证邮件。

> **注意** 此步骤可与步骤 1、2 **并行**执行。

3.1 验证域名

```
aws sesv2 create-email-identity \
  --email-identity 你的域名 \
  --region eu-south-2
```

3.2 获取 DKIM 记录

```
aws sesv2 get-email-identity \
  --email-identity 你的域名 \
  --query 'DkimAttributes.Tokens' \
  --output text \
  --region eu-south-2
```

输出 3 个 token（空格分隔）。

3.3 在 Cloudflare 添加 DKIM CNAME

对输出的每个 token，在 Cloudflare 添加一条 CNAME：

字段	值
Type	CNAME
Name	`{token}._domainkey`
Target	`{token}.dkim.amazonses.com`
Proxy	❌ DNS only

重复 3 次，每个 token 一条。

3.4 （可选）添加 SPF 记录

在 Cloudflare 添加 TXT 记录：

- **Name**: `@`
- **Value**: `v=spf1 include:amazonses.com ~all`

3.5 内测阶段：验证收件人邮箱

SES 沙箱模式只能给已验证邮箱发信。验证你自己的邮箱：

```
aws sesv2 create-email-identity \
  --email-identity your-email@example.com \
  --region eu-south-2
```

去邮箱点击验证链接。

TIP 内测阶段不需要申请移出沙箱。只要把测试用的收件人邮箱都验证一下就行。

第 4 步　部署主 CloudFormation 栈

这是核心步骤

一条命令创建所有 AWS 基础设施（约 50 个资源，耗时 15–25 分钟）。 确保步骤 2 的 ACM 证书已 `ISSUED`。

4.1 执行部署

```
ACM_CERTIFICATE_ARN="arn:aws:acm:eu-south-2:xxx:certificate/xxx" \
ALERT_EMAIL="你的告警邮箱" \
./deploy/deploy.sh setup
```

这会自动执行：

1. 构建 crab-auth Lambda zip（Docker 交叉编译 aarch64）
2. 上传 Lambda zip 到 S3
3. 构建 crab-cloud Docker 镜像
4. 推送到 ECR
5. 部署 CloudFormation 栈

4.2 创建的资源

资源	说明	月费
VPC	2 公有 + 2 私有子网, NAT Gateway	$35
RDS	PostgreSQL 16, db.t4g.micro, 加密, 14 天备份	$15
ECS Fargate	0.25 vCPU, 512MB (crab-cloud)	$10
ALB	HTTPS 443, TLS 1.3, WAF 关联	$18
NLB	TCP 8443, mTLS 透传	$18
Lambda	arm64, 256MB (crab-auth)	$1
WAF v2	限速 1000 req/5min + AWS 托管规则	含在 ALB
ECR + S3	镜像仓库 + 证书桶 (KMS 加密)	$1
Secrets Manager	4 个密钥	$1
CloudWatch	9 个告警 → SNS 邮件通知	$0
合计		**$99/月**

4.3 记录 Stack Outputs

部署完成后会输出关键信息。**务必记录**：

- ☐ **ALBDnsName**: `crab-xxx.eu-south-2.elb.amazonaws.com`
- ☐ **NLBDnsName**: `crab-mtls-xxx.elb.eu-south-2.amazonaws.com`
- ☐ **CrabAuthFunctionUrl**: `https://xxx.lambda-url.eu-south-2.on.aws/`
- ☐ **RDSEndpoint**: `crab-production.xxx.eu-south-2.rds.amazonaws.com`

如果忘了记录，可以再查：

```
aws cloudformation describe-stacks \
  --stack-name crab-production \
  --query 'Stacks[0].Outputs' \
  --output table \
  --region eu-south-2
```

第 5 步　配置 Secrets

5.1　获取 RDS 密码

CloudFormation 自动管理 RDS 密码。找到它：

```
# 找到 RDS 管理的 Secret 名称
aws secretsmanager list-secrets \
  --filter Key=name,Values=rds \
  --query 'SecretList[].Name' \
  --output text \
  --region eu-south-2
```

```
# 获取密码
aws secretsmanager get-secret-value \
  --secret-id 上面找到的secret名 \
  --query 'SecretString' \
  --output text \
  --region eu-south-2 | python3 -c "
import sys, json
d = json.load(sys.stdin)
print(f'postgres://{d[\"username\"]}:{d[\"password\"]}@RDS端点:5432/crab')
"
```

把输出中的 `RDS 端点` 替换为步骤 4 记录的 **RDSEndpoint**。

5.2　生成 JWT Secret

```
openssl rand -hex 32
```

✓ 记录输出的 64 位十六进制字符串。

5.3　运行 Secrets 脚本

```
./deploy/deploy.sh secrets
```

交互式输入 4 个值：

变量	值
database-url	postgres://crab:密码@RDS端点:5432/crab
stripe-secret-key	sk_test_...（内测用测试密钥）
stripe-webhook-secret	先输入占位值如 placeholder，步骤 7 再更新
jwt-secret	上面 openssl rand 生成的值

5.4 重启 ECS 使 Secrets 生效

```
aws ecs update-service \
  --cluster crab-production \
  --service crab-cloud \
  --force-new-deployment \
  --region eu-south-2 > /dev/null

echo "等待服务稳定..."
aws ecs wait services-stable \
  --cluster crab-production \
  --services crab-cloud \
  --region eu-south-2

echo "✓ 服务已就绪"
```

注意 crab-cloud 启动时会自动运行 PostgreSQL 迁移。如果 DATABASE_URL 正确，数据库 schema 会自动创建。

第 6 步 Cloudflare DNS 配置

用步骤 4 记录的 Stack Outputs 值。

6.1 添加 3 条 CNAME 记录

在 Cloudflare Dashboard → 你的域名 → **DNS** → **Add record**：

Type	Name	Target	Proxy
CNAME	`cloud`	步骤 4 的 ALBDnsName	❌ off
CNAME	`sync`	步骤 4 的 NLBDnsName	❌ off
CNAME	`auth`	步骤 4 的 CrabAuthFunctionUrl（去掉 `https://` 和尾部 `/`）	❌ off

所有记录的 Proxy 必须关闭（灰色云图标）！

- **`sync`**: **必须**关闭 — NLB TCP 透传 + mTLS，Cloudflare 代理会中断 TLS 握手
- **`cloud`**: 建议关闭 — ALB 已有 WAF + HTTPS，双重代理增加延迟、干扰 X-Forwarded-For
- **`auth`**: 建议关闭 — Lambda Function URL 自带 HTTPS

6.2 验证 DNS 解析

```
dig cloud.你的域名 +short
dig sync.你的域名 +short
dig auth.你的域名 +short
```

每个都应该返回对应的 AWS DNS 名称或 IP 地址。

TIP DNS 传播通常很快（Cloudflare 几秒钟），但偶尔需要等几分钟。

第 7 步 Stripe Webhook 配置

7.1 创建 Webhook Endpoint

Stripe Dashboard → **Developers** → **Webhooks** → **Add endpoint**:

Endpoint URL	`https://cloud.你的域名/stripe-webhook`
Version	Latest API version

订阅以下事件:

`checkout.session.completed`
`customer.subscription.created`
`customer.subscription.updated`
`customer.subscription.deleted`
`invoice.payment_failed`

7.2 获取 Signing Secret

创建完成后，点击该 endpoint → **Signing secret** → **Reveal**。

记录 `whsec_...` 值。

7.3 更新 Secrets Manager

```
aws secretsmanager put-secret-value \
  --secret-id "crab/production/stripe-webhook-secret" \
  --secret-string "whsec_你的签名密钥" \
  --region eu-south-2
```

7.4 重启 ECS 使新 Secret 生效

```
aws ecs update-service \
  --cluster crab-production \
  --service crab-cloud \
  --force-new-deployment \
  --region eu-south-2 > /dev/null
```

TIP ECS 新 Task 启动时会拉取最新的 Secrets Manager 值，旧 Task 会被自动替换。

第 8 步　冒烟测试

依次执行以下测试，确认所有服务正常。

8.1 Health Check

```
curl -s https://cloud.你的域名/health
```

✓ 期望: HTTP 200，返回健康状态

8.2 Lambda 响应

```
curl -s https://auth.你的域名/
```

✓ 期望: 非 5xx 响应

8.3 ECS 日志检查

```
aws logs tail /ecs/crab-cloud-production \
  --since 10m \
  --region eu-south-2
```

✓ 期望: 看到启动日志，含 migration 相关信息

8.4 部署状态全览

```
./deploy/deploy.sh status
```

✓ ECS: Running = 1, Desired = 1 ✓ Lambda: State = Active ✓ RDS: Status = available ✓ Active Alarms: None

8.5 注册测试租户

```
curl -s -X POST https://cloud.你的域名/register \
  -H "Content-Type: application/json" \
  -d '{
    "email": "你验证过的邮箱",
    "password": "TestPassword123!",
    "restaurant_name": "测试餐厅"
  }' | python3 -m json.tool
```

✓ 期望: 返回注册成功的 JSON 响应

8.6 验证邮件

检查邮箱是否收到验证邮件。

> **注意** SES 沙箱模式下，收件人邮箱必须在步骤 3 中已验证。

8.7 Stripe Webhook 测试

安装 Stripe CLI（如未安装）：

```
brew install stripe/stripe-cli/stripe
stripe login
```

监听并转发：

```
stripe listen --forward-to https://cloud.你的域名/stripe-webhook
```

在另一个终端触发测试事件：

```
stripe trigger checkout.session.completed
```

✓ 期望: stripe listen 输出显示事件已成功处理 (200)

第 9 步 SNS 告警确认

CloudFormation 创建了 SNS 订阅，AWS 会发送确认邮件到你的告警邮箱。

9.1 确认订阅

1. 打开步骤 4 中 `ALERT_EMAIL` 指定的邮箱
2. 找到来自 AWS 的确认邮件
3. 点击 **Confirm subscription** 链接

9.2 验证

```
TOPIC_ARN=$(aws cloudformation describe-stacks \
  --stack-name crab-production \
  --query 'Stacks[0].Outputs[?OutputKey==`AlarmTopicArn`].OutputValue' \
  --output text \
  --region eu-south-2)

aws sns list-subscriptions-by-topic \
  --topic-arn "$TOPIC_ARN" \
  --query 'Subscriptions[].{Endpoint:Endpoint,Status:SubscriptionArn}' \
  --output table \
  --region eu-south-2
```

✓ 期望: Status 不是 `PendingConfirmation`，而是一个完整的 ARN

完成检查清单

所有步骤完成后，逐项确认：

步骤	检查项	状态
1	GitHub OIDC Role 已创建，GitHub Secrets 已配置	☐
2	ACM 证书状态 = ISSUED	☐
3	SES 域名已验证，收件人邮箱已验证	☐
4	CloudFormation 栈状态 = CREATE_COMPLETE	☐
5	Secrets Manager 4 个密钥都有真实值	☐
6	3 条 DNS CNAME 已添加，Proxy 全部关闭	☐
7	Stripe Webhook endpoint 已创建，Secret 已更新	☐
8	Health check 返回 200，注册流程正常	☐
9	SNS 告警邮件已确认	☐

恭喜！内测环境已就绪。

现在可以：

- 向 main 分支 push 代码，CI/CD 自动构建部署
- 注册测试租户，验证完整流程
- 在餐厅安装 edge-server + POS 客户端，测试激活和同步

后续待办（非紧急）：

- 申请 SES Production Access（移出沙箱）
- 配置 Tauri 签名密钥 + Release 流水线
- 创建 CloudWatch Dashboard
- 编写运维 Runbook