



云原生社区 meetup  
第七期·深圳站

# 使用IAST构建高效的 DevSecOps流程

董志勇（火线平台-洞态产品负责人）





# 为什么要做安全测试？

## 国家法律法规的出台

《中华人民共和国网络安全法》  
《中华人民共和国网络隐私保护法》  
《中华人民共和国数据安全法》  
《中华人民共和国个人信息保护法》

...

# 为什么要做安全测试？

- 勒索软件：2017年“永恒之蓝”
- 数据泄漏：2020年以色列640万选民数据泄漏
- 服务器被入侵等等

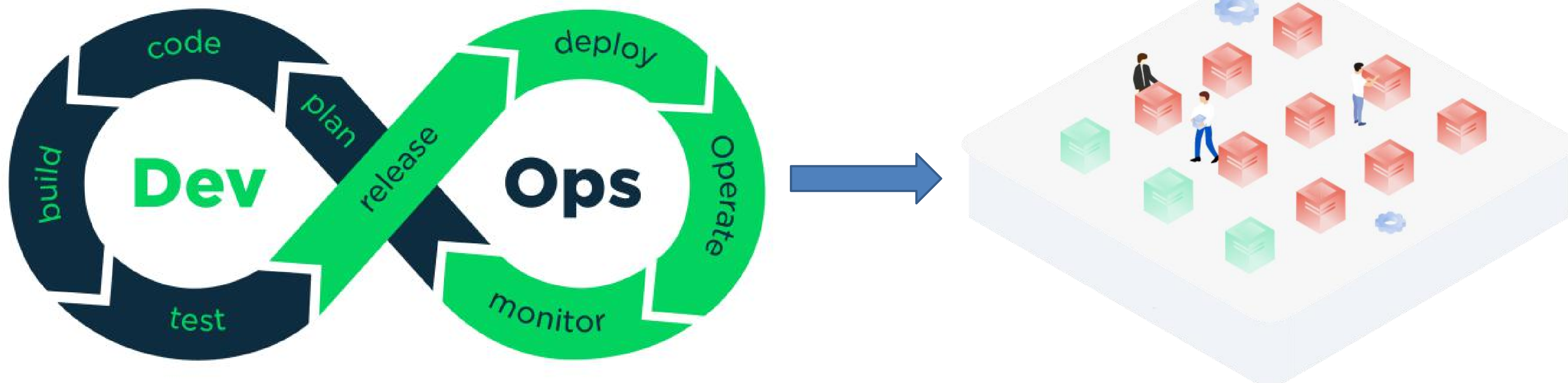
## 网络攻击频发

# DevOps 前的安全测试



迭代速度慢、安全测试任务密度小

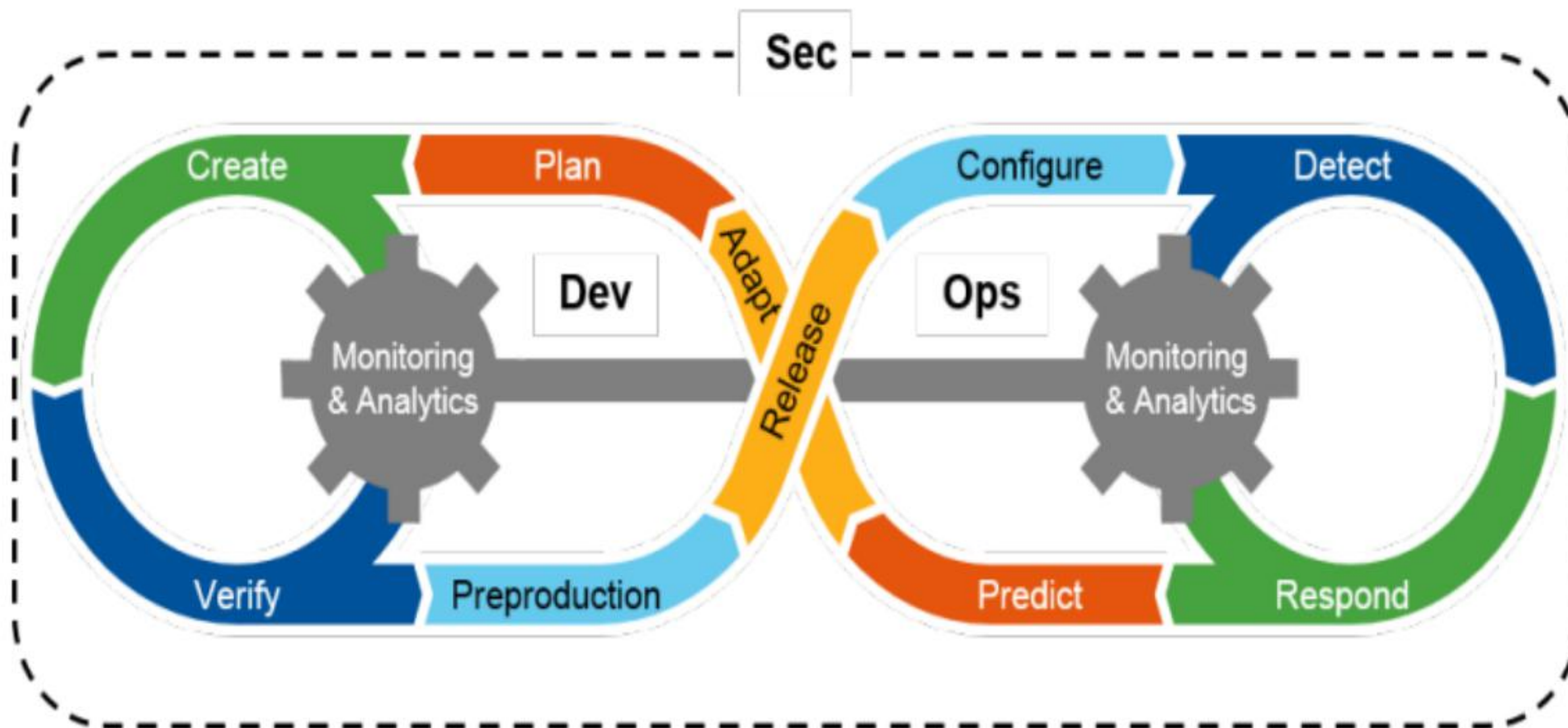
## DevOps 带来的新挑战



DevOps提高了企业的生产效率，实现了应用的快速迭代，但带来了待上线应用的数量与安全测试人员数量的严重不对等。为了满足企业的生产效率需求，必须有同样高效的安全工具来解决安全测试人员数量不足给生产效率带来的影响。



# DevSecOps



Source: Gartner (September 2016)

[security.tencent.com](https://security.tencent.com)

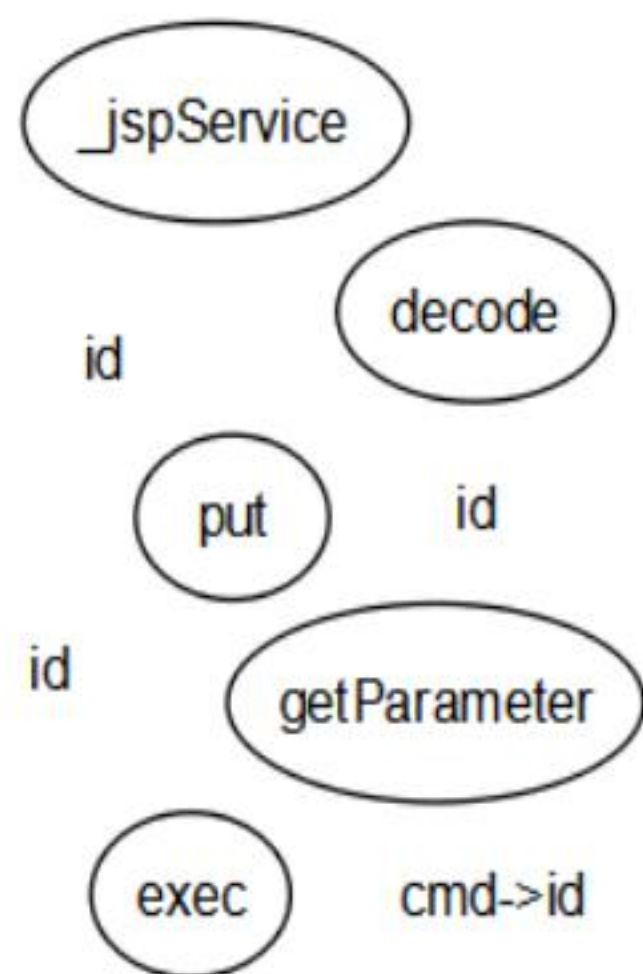
# 什么是 IAST 呢？

IAST全称Interactive Application Security Testing, 译为：“交互式应用程序安全测试”。

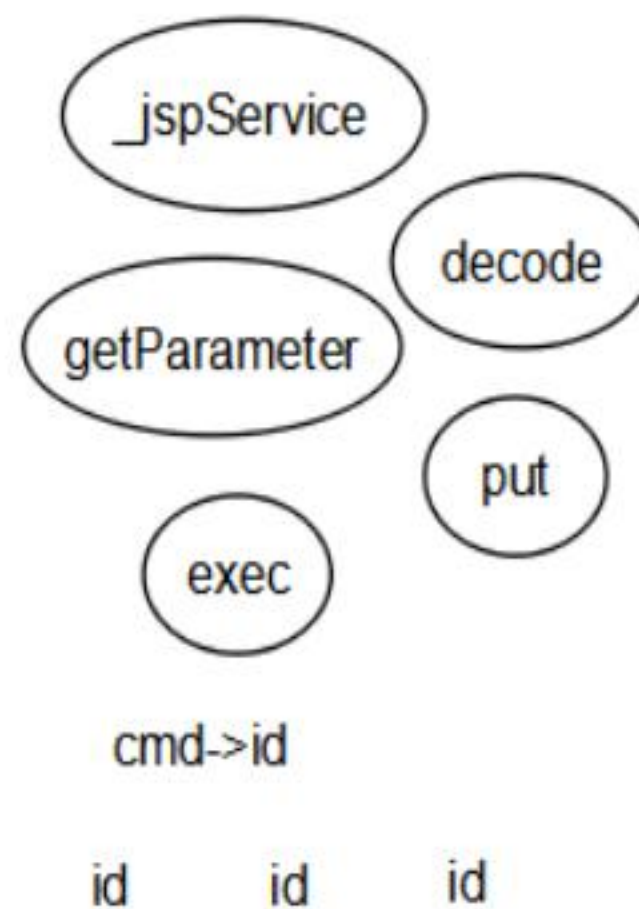
它通过Agent来监控应用程序运行时的函数执行情况，采集相关数据，并与服务端进行实时交互，从而**高效、准确的识别出应用程序中的漏洞**，同时可准确的**定位到漏洞所在的文件、行数、方法及参数**，方便开发团队及时的**修复漏洞**。

# I A S T 检测原理

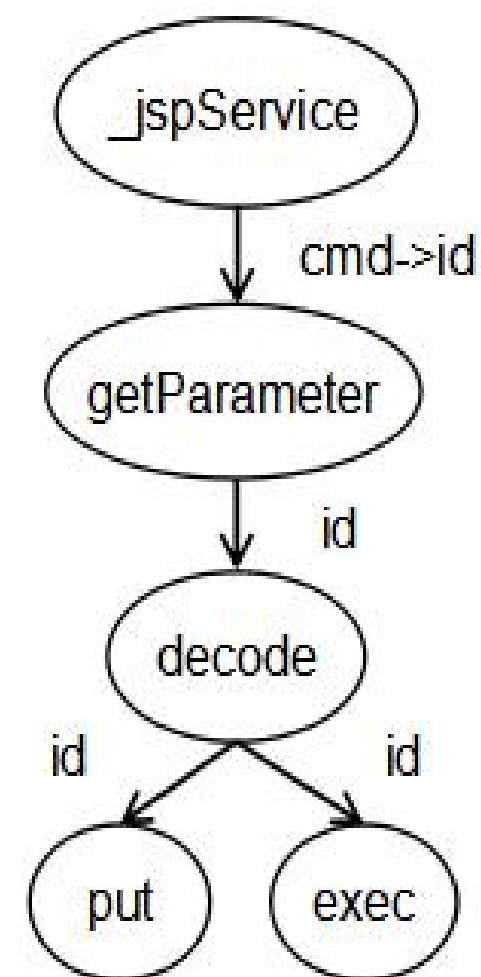
不可信数据采集



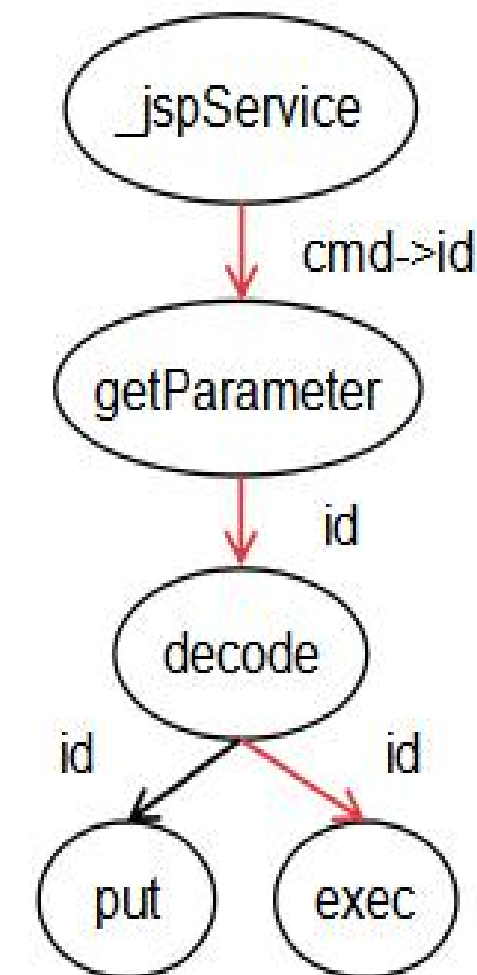
不可信数据预处理



不可信数据传播图

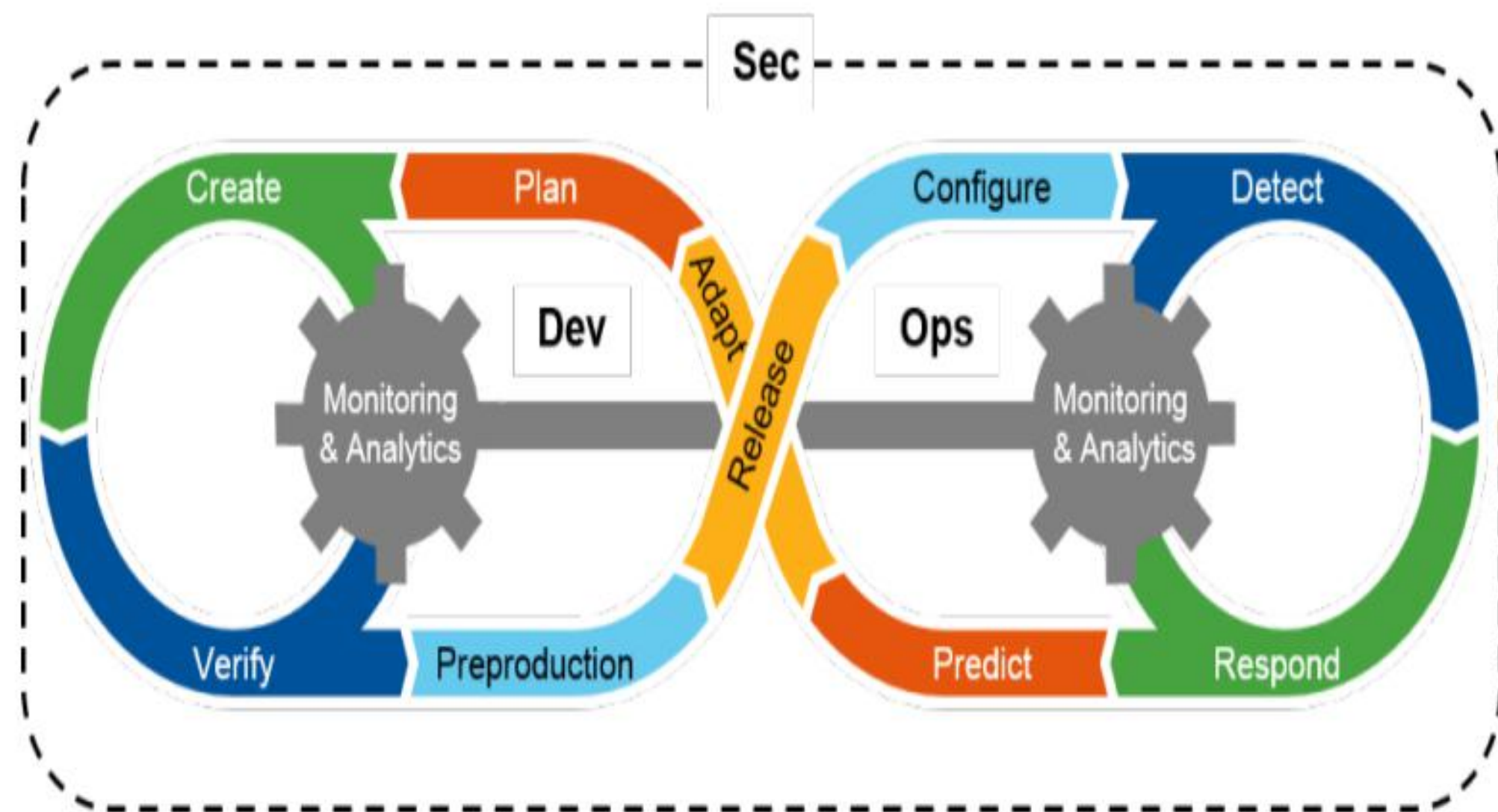


数据调用链路查找



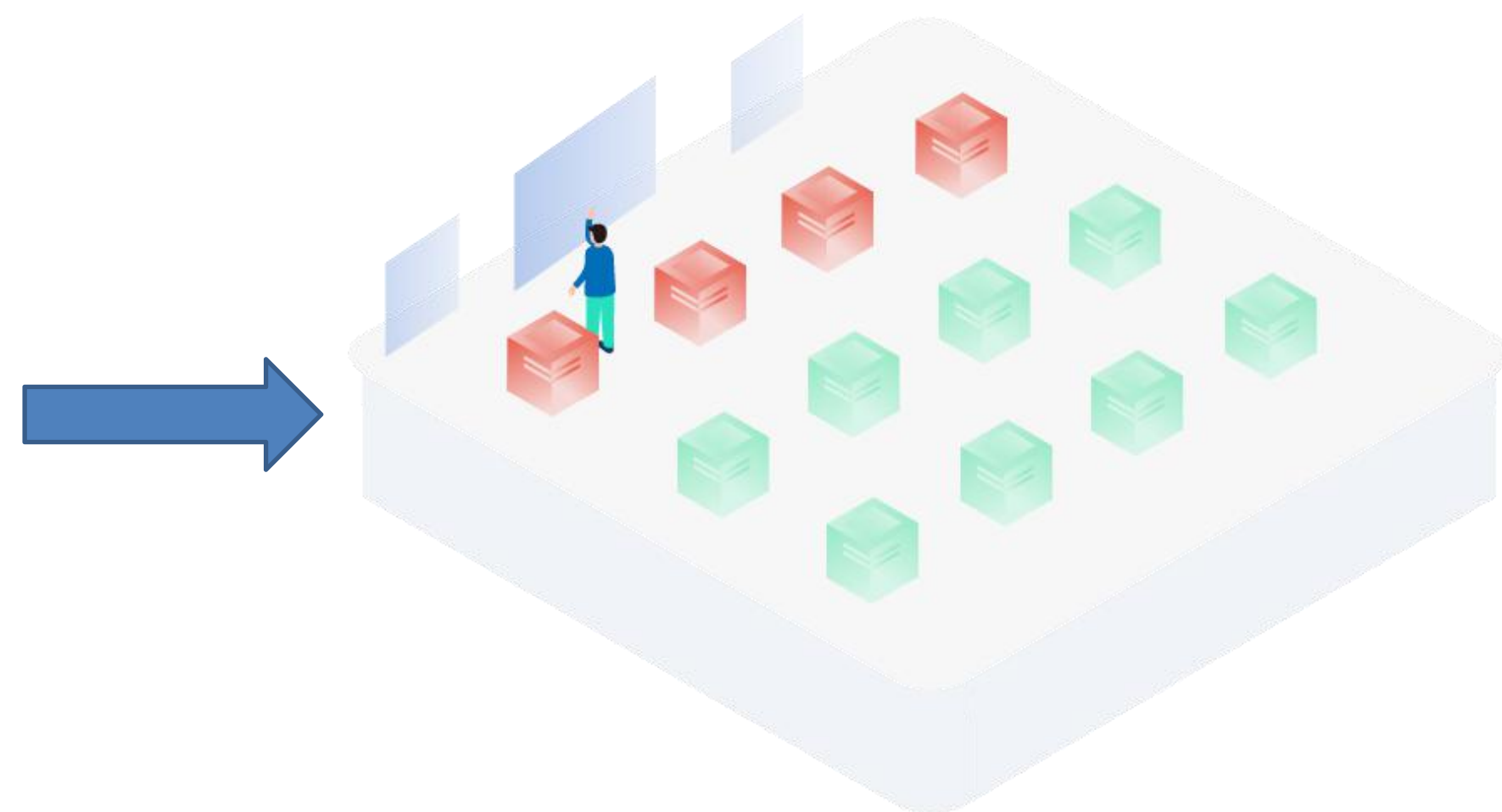


# I A S T + D e v S e c O p s



Source: Gartner (September 2016)

[security.tencent.com](https://security.tencent.com)



# 为什么选择洞态 IAST



为什么选择洞态 IAST

# 依赖组件的供应链风险检查



# 为什么选择洞态 IAST

- 传统WEB应用
- 前后端分离场景下的漏洞检测
- 验证码场景下的漏洞检测
- 数据包加密场景
- 防重放签名等场景
- 分布式架构下的漏洞检测
- 微服务架构下的漏洞检测

## 支持各种业务场景

# 为什么选择洞态 IAST

- 类的API Sitemap，方便各部Swagger门人员查看
- 提供测试覆盖率，精确的反馈出未测试到的接口
- 准确的参数名称及数据类型，可用于直接发送HTTP请求，提高接口覆盖率

The screenshot displays the Tongtai IAST web interface. At the top, there's a navigation bar with tabs: 项目配置 (Project Configuration), 应用漏洞 (Application Vulnerabilities), 组件管理 (Component Management), 搜索 (Search), 系统配置 (System Configuration), 组织管理 (Organization Management), and 租户管理 (Tenant Management). A '+ Add Agent' button and a flame icon are on the right.

The main content area shows a project named 'springsec' with a 'JAVA' tag. Below this, there's a status bar with: 扫描模式 插桩模式 (Scan Mode: Instrumentation Mode), 负责人 test002 (Responsible: test002), 最新时间 2021.08.24 12:37:44 (Latest Time: 2021.08.24 12:37:44), 版本 0824 (Version: 0824), and buttons for 报告导出 (Export Report) and 设置 (Settings).

Below the status bar, there are four tabs: 项目概况 (Project Overview), 项目漏洞 (Project Vulnerabilities), 项目组件 (Project Components), and API导航 (API Navigation). The API Navigation tab is active, showing a search bar with '请选择请求方法' (Select Request Method), '请选择覆盖状态' (Select Coverage Status), and '请输入API地址进行搜索' (Enter API address to search). The coverage rate is 13.27%.

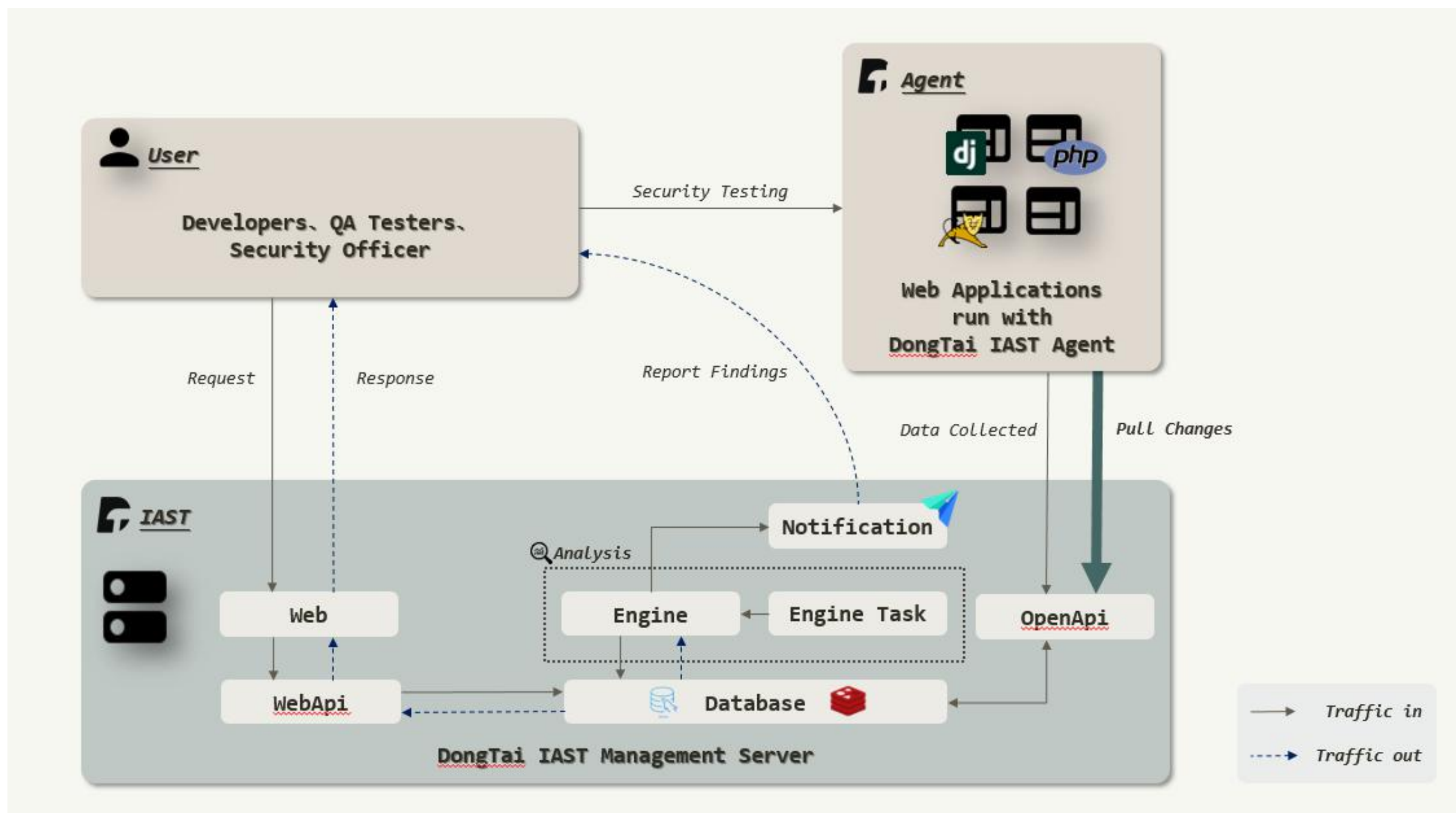
The API list shows two entries:

- GET /vul/cmd-002 (Status: Green checkmark, Command Execution button). Below it is a table with parameters:

名称	类型	额外信息
arg0	Map	GET请求参数
响应	String	

- GET /vul/cmd-003/{cmd} (Status: Green checkmark, Command Execution button).

# 为什么选择洞态 IAST



统一探针上报的数据格式、构建数据底座、利用数据分析实现漏洞的离线检测



# 为什么选择洞态 IAST



项目配置

应用漏洞

组件管理

搜索

系统配置

组织管理

租户管理

+ Add Agent



筛选



请输入查询内容



常用查询语法：

URL `(.*)/druid/.*`

请求体 `(.*)whoami(.*)`

方法签名 `whoami`

响应头 `set-cookie`

请求头 `(.*)exec`

请求体 `<script> alert(1) </script>`

污点数据 `(.*)rememberMe(.*)`

# 为什么选择洞态 IAST

洞态

项目配置应用漏洞组件管理搜索系统配置组织管理租户管理

+ Add Agent

方法签名Runtime.exec

http://localhost:8080/cmd

探针: Mac OS X-iliusky-v1.0.0-61862e3851934b9d9... 用户: demo 关联漏洞: 命令执行

HTTP数据包污点调用链

GET /cmd?cmd=ls HTTP/1.1  
host:localhost:8080  
connection:keep-alive  
cache-control:max-age=0  
sec-ch-ua:"Google Chrome";v="93", " Not;A Brand";v="99",  
"Chromium";v="93"  
sec-ch-ua-mobile:?0  
sec-ch-ua-platform:"macOS"  
upgrade-insecure-requests:1  
user-agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7)

HTTP/1.1 200  
DongTai:v1.0.3

# 为什么选择洞态 IAST

洞态

项目配置应用漏洞组件管理搜索系统配置组织管理租户管理

+ Add Agent

系统配置

引擎管理

规则总数: 584 条规则类型: 66 种sink规则数量: 309 条

污点源方法规则传播方法规则过滤方法规则危险方法规则

规则集 危险方法规则

规则类型 Sql注入

规则详情 java.sql.Statement.execute(java.lang.String)

污点来源 参数 1 增加

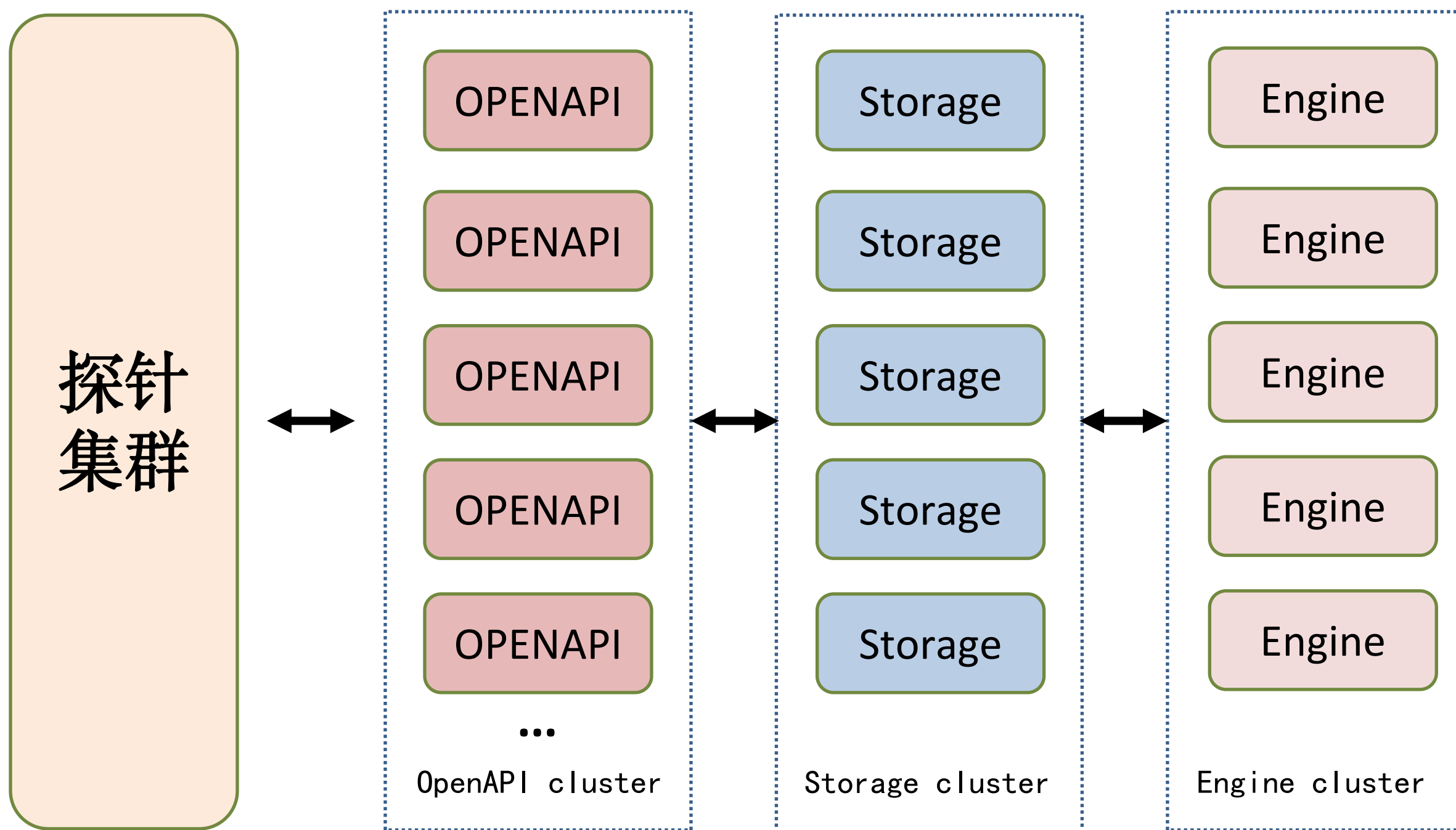
污点追踪 ☒ 启用 ☐ 禁用

继承深度 ☐ 仅当前类 ☒ 仅子类 ☐ 当前类及子类

取消 确定



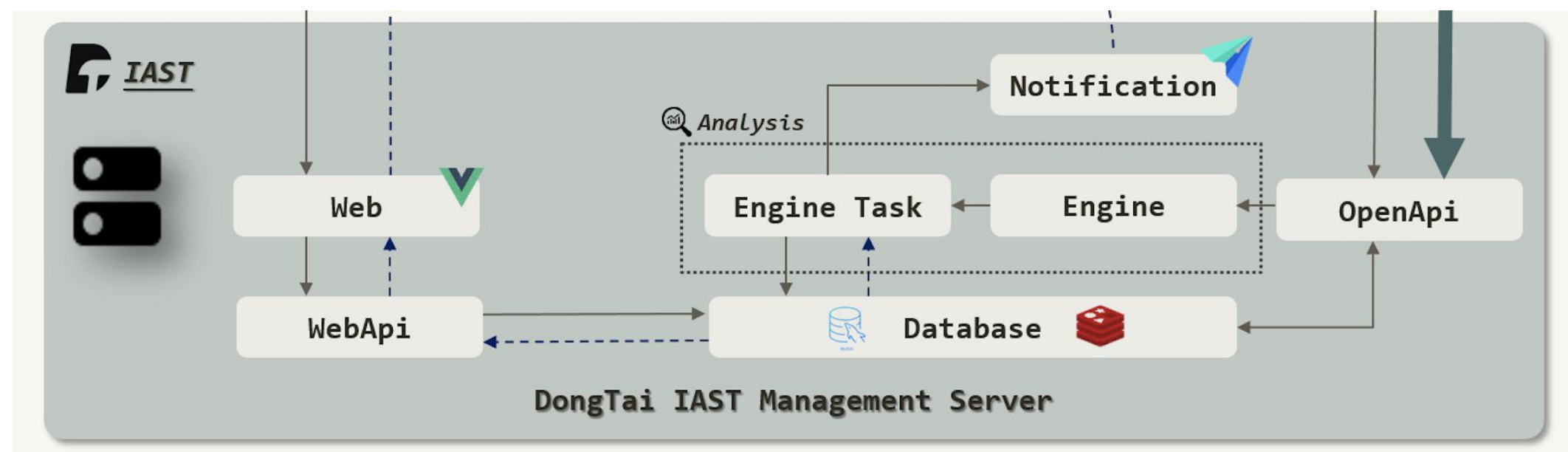
# 洞态 IAST + Kubernetes



无状态、弹性扩容、自动伸缩

# 洞态 IAST + Kubernetes

```
# dongtai-webapi服务
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dongtai-webapi
  namespace: CHANGE_THIS_NAMESPACE
  annotations:
    kubesphere.io/description: dongtai-webapi接口
  labels:
    app: dongtai-webapi
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dongtai-webapi
  template:
    metadata:
      labels:
        app: dongtai-webapi
    spec:
      containers:
        - name: dongtai-webapi-container
          image: registry.cn-beijing.aliyuncs.com/huoxian_pub/dongtai-webapi:latest
          imagePullPolicy: IfNotPresent
```



# 洞 态 I A S T   A g e n t   +   D o c k e r

```
FROM huoxian_pub/dongtai-java-agent:1.0.4-jdk8
```

```
# ... build your java application
```



# 洞态 IAST Agent + Kubernetes

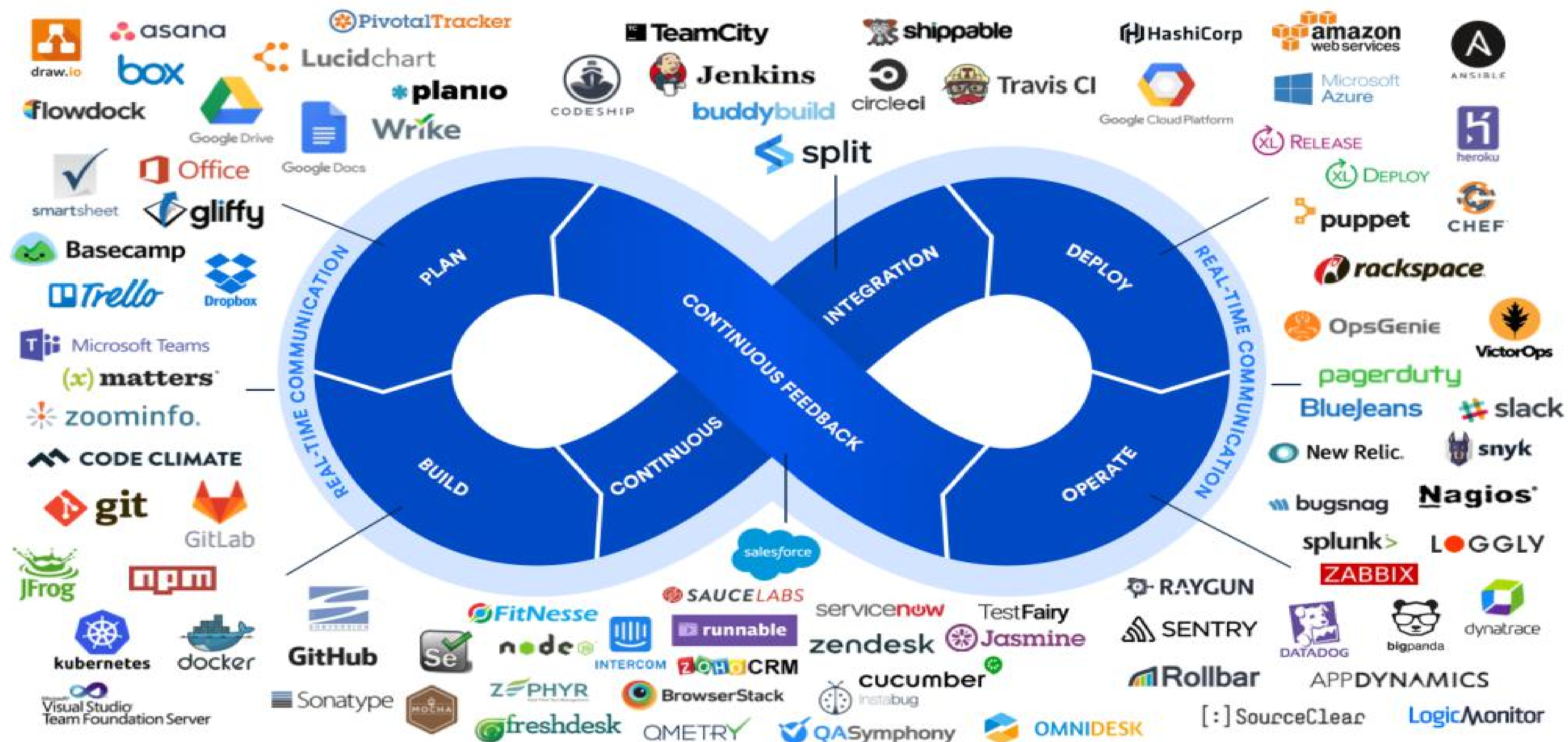
```
apiVersion: v1
kind: Pod
metadata:
  name: agent-as-sidecar
spec:
  restartPolicy: Never

  volumes:
  - name: skywalking-agent
    emptyDir: {}

  containers:
  - name: agent-container
    image: huoxian_pub/dongtai-java-agent:1.0.4-alpine
    volumeMounts:
    - name: dongtai-agent
      mountPath: /agent
    command: ["/bin/sh"]
    args: ["-c", "cp -R /dongtai/agent /agent/"]

  - name: app-container
    image: apache/Spring-Boot-Example:latest
    volumeMounts:
    - name: dongtai-agent
      mountPath: /agent
    env:
    - name: JAVA_TOOL_OPTIONS
      value: "-javaagent:/dongtai/agent/agent.jar"
```

# 洞态 I A S T + D e v O p s





想了解更多，请关注我们的公众号或点击

<https://cloudnative.to>



云原生社区

Cloud Native Community