



GOPS2018  
Shenzhen

# GOPS

全球运维大会 2018

2018.4.13-4.14

中国·广东·深圳·南山区 圣淘沙大酒店(翡翠店)





GOPS2018  
Shenzhen

# Python开发36计之企业级运维平台开发实战

郭宏泽 技术总监



GOPS2018  
Shenzhen

# 目录

**1** 平台规划

**2** 技术细节

**3** 功能模块

**4** 开发心得

# 企业运维难点



GOPS2018  
Shenzhen

## 系统架构

随着IAAS云化，容器化，PAAS智能化，依赖库多样化，RUNTIME复杂化，运维人员无法应对新时代的运维要求。

## 技术债务

由于资源成本压力，中小企业的IT运维及IT系统存在大量历史债务，其结果就是在产品和服务上无法满足需求，形成恶性循环

## 人员流动

由于系统的混乱，高层不懂技术，导致牛人招请不起，技术好的招不来，来了待不久，技术人员流动率大，系统稳定性，可持续性极差。

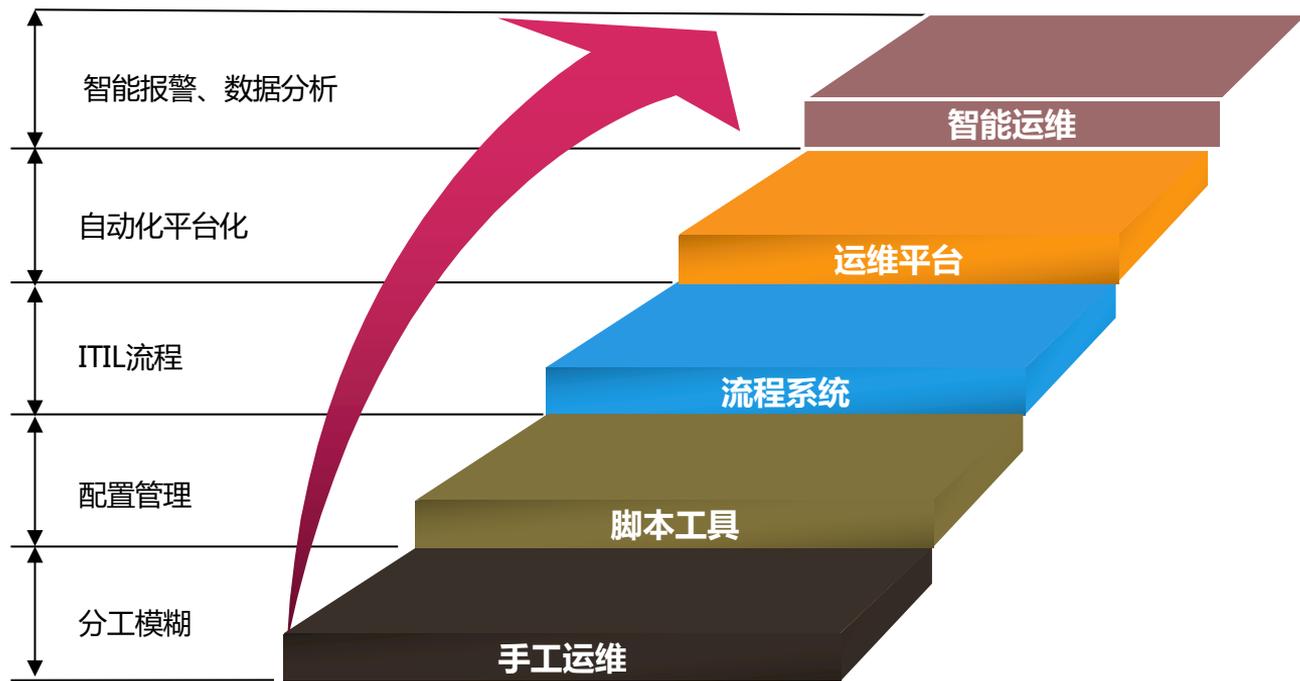
## 战略规划

大多中小企业没有IT战略规划，在IT资源投入上分配不合理，该花钱的没花，不该花的浪费严重，技术团队无法支撑企业IT的实际要求。

# 运维架构的演进



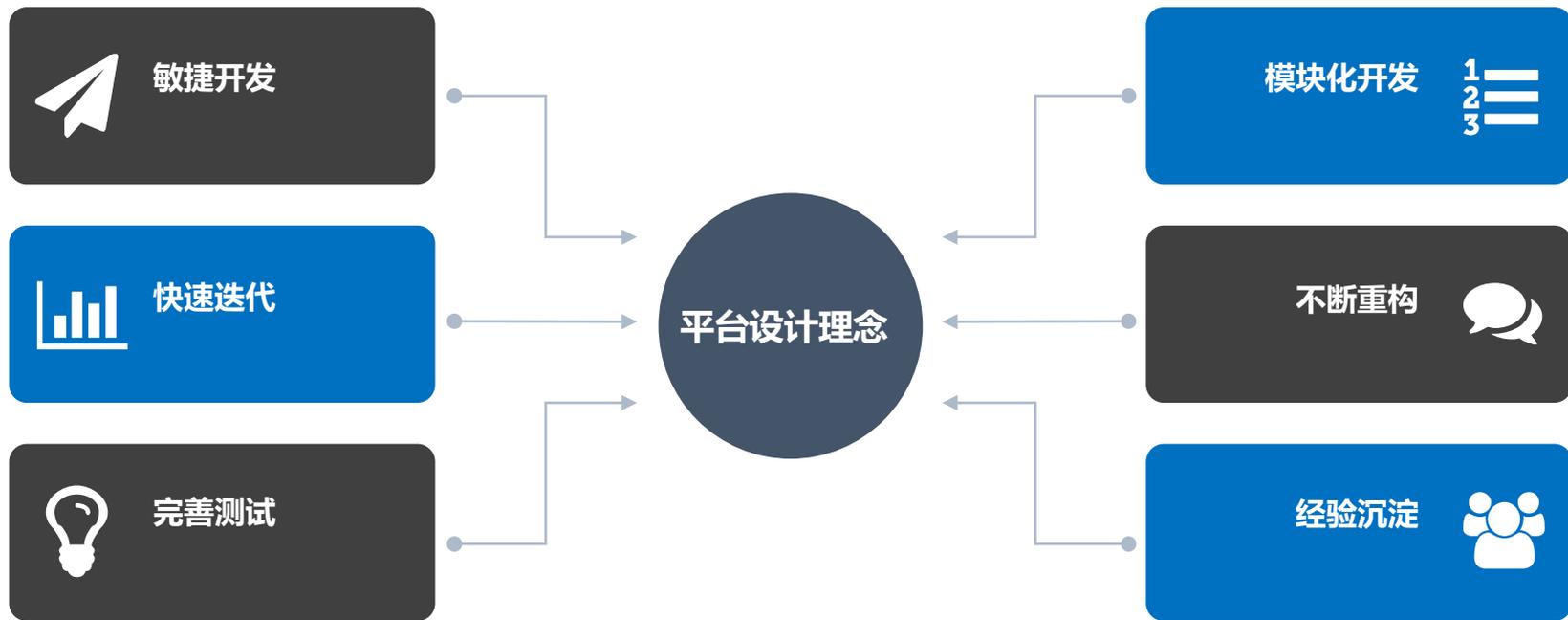
GOPS2018  
Shenzhen



# 运维平台开发理念



GOPS2018  
Shenzhen





## 运维模块

部署自动化  
配置自动化  
巡检自动化  
作业管理  
故障自愈  
部署流水线

## 监控模块

基础监控  
系统监控  
应用监控  
业务监控  
服务监控  
终端监控

## 安全模块

准入管理  
用户管理  
权限管理  
角色管理  
授权管理  
操作审计

## 流程模块

事件管理  
变更管理  
资源管理  
发布管理  
知识管理  
流程规范

## 智能模块

数据分析  
事件分析  
性能分析  
趋势预测  
智能决策  
ChatOps

## 运营模块

设备采购  
合同管理  
成本管理  
预算管理  
运营报表  
考核管理

## 运维平台API

## 应用配置

软件包

配置文件

监控属性

业务属性

架构属性

权限属性

运营属性

## 资产管理

IDC

Network

Server

虚拟化

CDN

操作系统

DNS



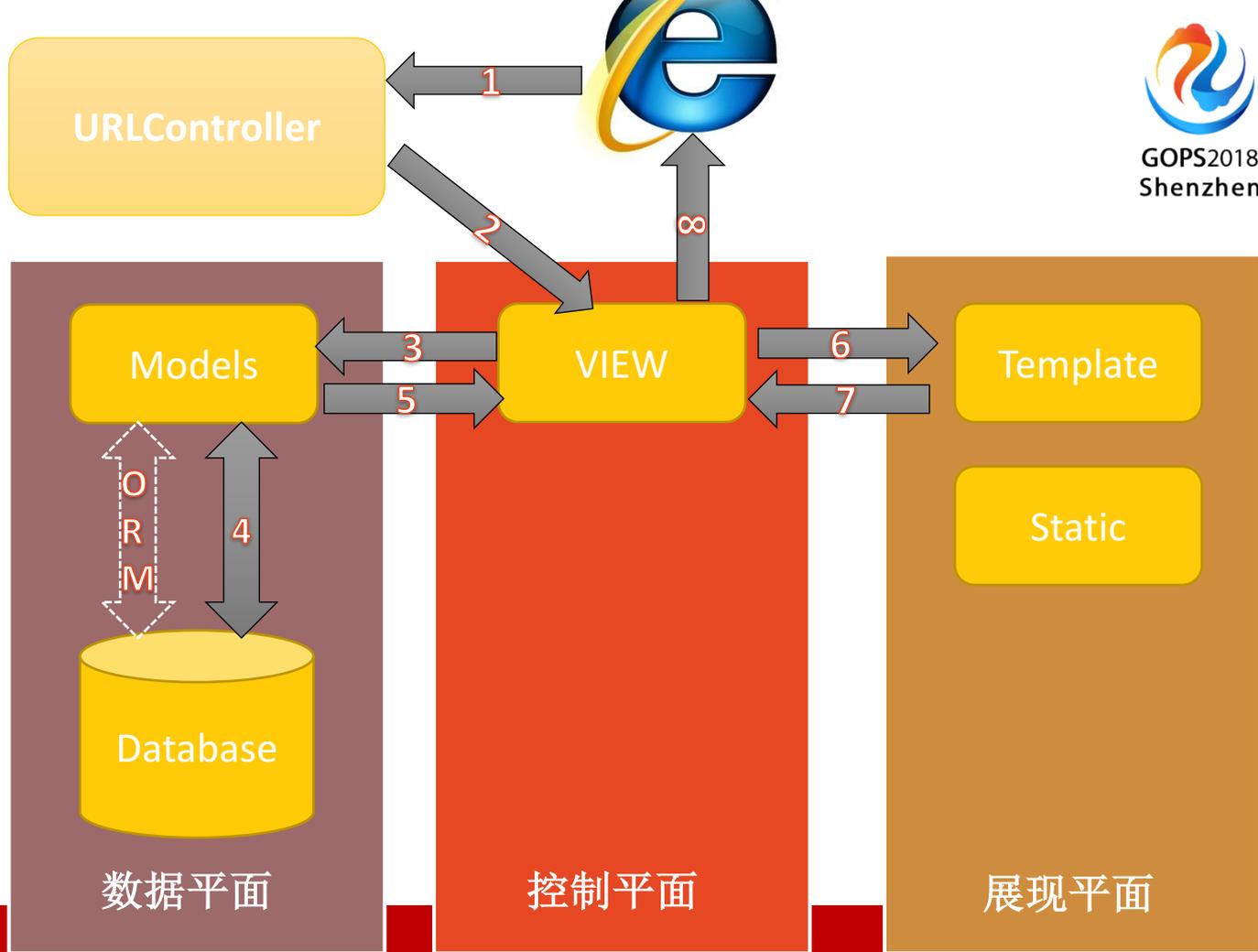
GOPS2018  
Shenzhen

# ◆ Python+ Django+ Mysql+ Redis+ MongoDB





- 1、Requests
- 2、Forward
- 3、Call
- 4、ORM
- 5、Return Data
- 6、Templete
- 7、Render
- 8、Response





GOPS2018  
Shenzhen

# 前端资源

- ◆ Adminlte 主体框架
- ◆ Bootstrap 图形条目
- ◆ Jquery 特效应用
- ◆ font-awesome 图标字体
- ◆ Layer 弹窗组件
- ◆ Echarts 监控图表

# 资产管理



GOPS2018  
Shenzhen



adm

AdminSet



主机管理

机房管理

机柜管理

属组管理

站点导航

资产管理

应用管理

任务编排

持续交付

监控平台

用户管理

系统配置

添加设备

机房

主机组

设备类型

设备状态

Search

搜索

导出

Show 10 entries

<input type="checkbox"/>	主机名	管理IP	属组	CPU	内存(G)	机房	功能
<input type="checkbox"/>	bj-zd2f-vm-xmzj-prod-redis3	172.30.9.45		4	8	中电二层	WebSSH 编辑 删除
<input type="checkbox"/>	bj-zd2f-ps-node1	172.30.0.2		48	126	中电二层	WebSSH 编辑 删除
<input type="checkbox"/>	bj-zd2f-vm-xmzj-prod-web2	172.30.9.49		4	8	中电二层	WebSSH 编辑 删除
<input type="checkbox"/>	bj-zd2f-vm-xmzj-prod-mysql2	172.30.9.47		4	8	中电二层	WebSSH 编辑 删除
<input type="checkbox"/>	bj-zd2f-vm-xmzj-test-redis1	172.30.9.50		4	8	中电二层	WebSSH 编辑 删除
<input type="checkbox"/>	bj-zd2f-vm-xmzj-test-cassandra1	172.30.9.52		4	8	中电二层	WebSSH 编辑 删除



GOPS2018  
Shenzhen

# CMDB表结构设计

- ✓ Host 服务器
- ✓ Group 服务器组
- ✓ IDC 所在机房
- ✓ Cabinet 机柜
- ✓ Asset\_type 资产类型
- ✓ Asset\_status 资产状态
- ✓ 产品线
- ✓ 工程
- ✓ 负责人
- ✓ 认证中心



GOPS2018  
Shenzhen

# 基于CMDB的应用配置信息

程序部署路径:

配置文件路径:

所属产品线:

项目负责人:

所在服务器:

可选择服务器

Showing all 1

Filter

→ →	→
BJ-ZD-L2-ADS1	

已选择服务器

Showing all 2

Filter

←	← ←
adminset	
jumpserver	

# CMDB整合ANSIBLE



GOPS2018  
enzhen

同步数据

选择主机

x cn-bj-web02 x cn-bj-web01 x cn-bj-web03

选主机组

选择组(group)

Roles

Playbook Roles

Playbook

Playbook

执行

Ansible Command

use enter spilt and command like: ansible web -a 'ls'

执行



# Agent上报接口

- ✓ 使用Django HTTP SERVER而非Socket
- ✓ 使用Http Post方法上传
- ✓ 使用Json传递数据
- ✓ 数据处理后直接插入数据库

```
@token_verify()
```

```
def collect(request):
```

```
    asset_info = json.loads(request.body)
```

```
    if request.method == 'POST':
```

```
        vendor = asset_info['vendor']
```

```
        disk = asset_info['disk']
```

```
        cpu_model = asset_info['cpu_model']
```

```
host.save()
```



GOPS2018  
Shenzhen

# Agent设计原则

- ✓ Python 2.7兼容3.x语法
- ✓ Shell直接提取数据
- ✓ Psutil数据采集
- ✓ 其它数据采集 ( SNMP )
- ✓ 使用Http Post Json数据到服务器API接口
- ✓ Schedule 循环采集
- ✓ 使用Systemd或supervisor进程守护



# Json上报监控信息

```
{
  'mem': {
    'available': 2798,
    'used': 764,
    .....
  },
  'hostname': 'adminset',
  'token': 'eMRdmYHXcb3P',
  'net': [{
    'nic_name': 'lo',
    'traffic_out': 5,
    'traffic_in': 5
  }, {
  },
  'disk': [{
    'mountpoint': '/',
    'total': 57.78,
    'used': 5.12,
    'free': 52.66,
    'percent': 8.9
  },
  'cpu': {
    'softirq': 0.0,
    'percent': 2.5,
    'lcpu_percent': [1.0, 5.0],
    'system': 0.0,
  }
}
```



2018  
zhen

```
def token_verify():

    def decorator(view_func):
        def _wrapped_view(request, *args, **kwargs):
            set_token = get_dir('token')
            error_info = "Post forbidden, your token error!!"
            if request.method == 'POST':
                post_token = json.loads(request.body)
                if set_token == post_token["token"]:
                    return view_func(request, *args, **kwargs)
                else:
                    return HttpResponse(error_info, status=403)

            return HttpResponse(error_info, status=403)

        return _wrapped_view
```

# 任务调度中心



GOPS2018  
Shenzhen

定时任务 Ansible Shell adm

任务列表 Interval Crontab 任务结果 后台管理

新建任务

Show 10 entries Search:

<input type="checkbox"/>	任务名称	任务类型	任务参数	上次运行	运行次数	上次修改	过期时间
<input type="checkbox"/>	<a href="#">celery.backend_cleanup</a>	celery.backend_cleanup	[]	June 23, 2017, 4:06 p.m.	8	June 23, 2017, 6:04 p.m.	None
<input type="checkbox"/>	<a href="#">ghz100</a>	setup.tasks.command	[]	None	750	June 23, 2017, 6:04 p.m.	None
<input type="checkbox"/>	<a href="#">test-script</a>	setup.tasks.script	[]	None	88	May 25, 2017, 10:06 p.m.	None

Showing 1 to 3 of 3 entries

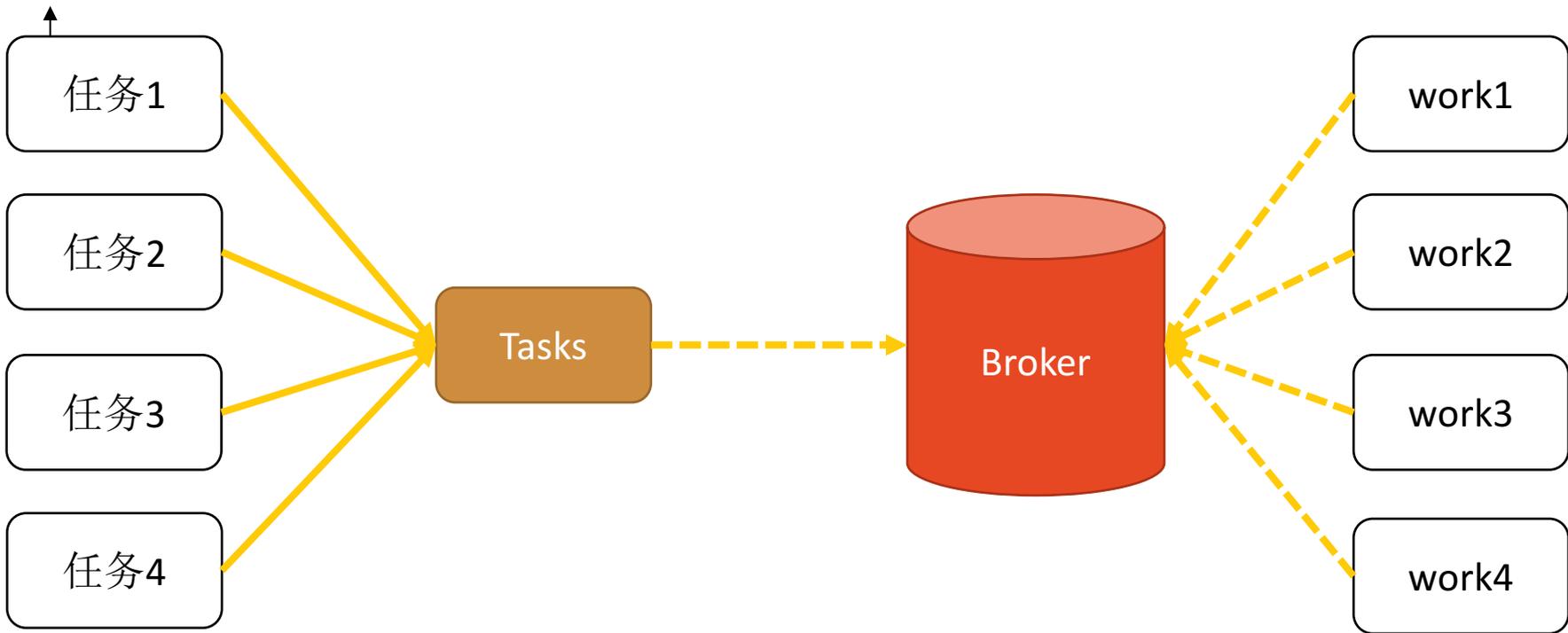
Previous 1 Next

删除

# 异步任务处理



GOPS2018  
Shenzhen





GOPS2018  
Shenzhen

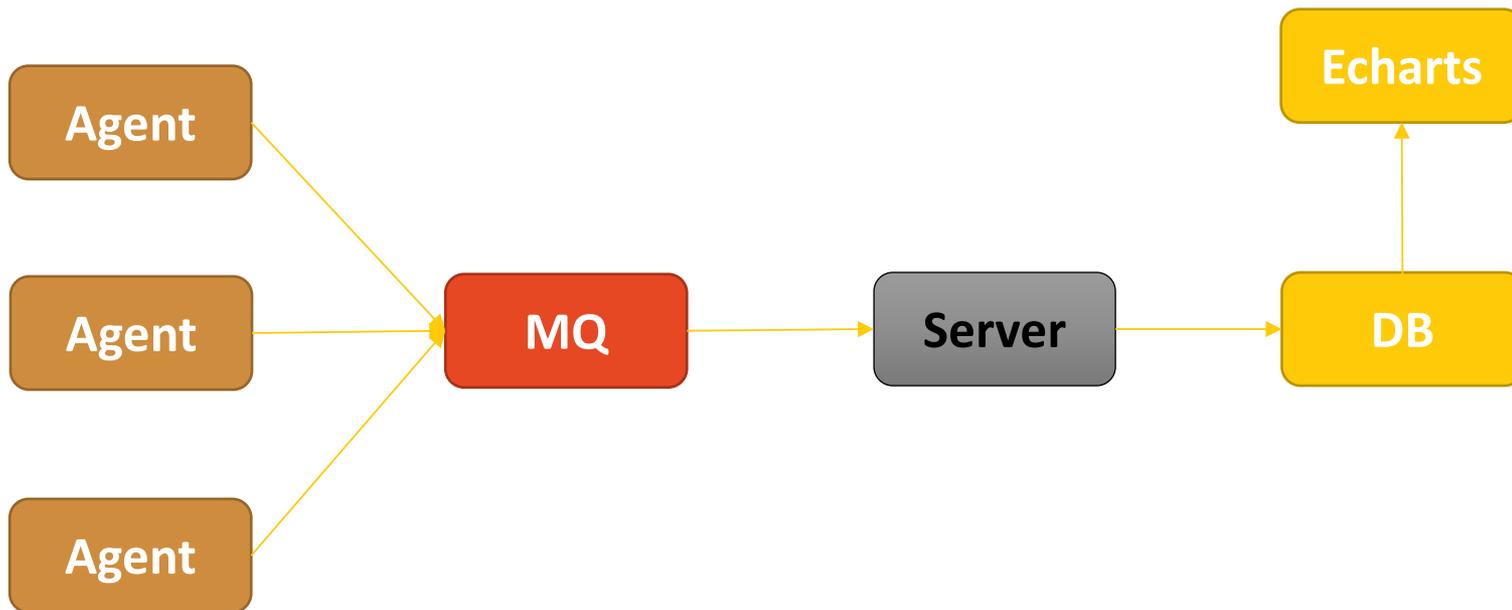
# 编写监控模块

- ✓ 不要使用关系型数据库存储监控数据
- ✓ 推荐使用influxdb rrdtool prometheus存储监控数据
- ✓ 本例使用mongodb进行存储监控数据
- ✓ 使用Echarts或Grafana图表展现
- ✓ 使用Json数据传递
- ✓ 使用JS异步回调实时刷新

# Agent上报模式



GOPS2018  
Shenzhen



# 自动发现的监控平台



GOPS2018  
Shenzhen



# 持续交付



GOPS2018  
Shenzhen

AdminSet ☰

- 站点导航
- 资产管理
- 应用管理
- 任务编排
- 持续交付**
- 监控平台
- 用户管理
- 系统配置

[添加项目](#)

Show  entries Search:

<input type="checkbox"/>	Name	Status	Progress	Policy	Action
<input type="checkbox"/>	web	<span style="color: blue;">●</span>	<div style="width: 20%;"></div>	Direct	<a href="#">部署</a> <a href="#">日志</a> <a href="#">编辑</a> <a href="#">删除</a>

Showing 1 to 1 of 1 entries

[Previous](#) **1** [Next](#)

[删除项目](#)



GOPS2018  
Shenzhen

# AdminSet

- 基于Python Django1.11开发的自动化运维平台，主要功能：资产管理、任务编排、持续交付、系统监控、运维管理、配置管理。
- Github
- <https://github.com/guohongze/adminset>



GOPS2018  
Shenzhen



# Thanks

高效运维社区  
开放运维联盟

荣誉出品



GOPS2018  
Shenzhen

想第一时间看到高效运维社区  
的新动态吗？

