

DevOps在重大活动网站安全保障中的应用

厦门大学 郑海山 <https://dog.xmu.edu.cn>

2017/11/24 武汉大学

// CONTENT //

01

静态网站安全性

JINGTAI WANGZHAN ANQUAN XIN

02

DevOps介绍

SUIBIAN LUANXIE

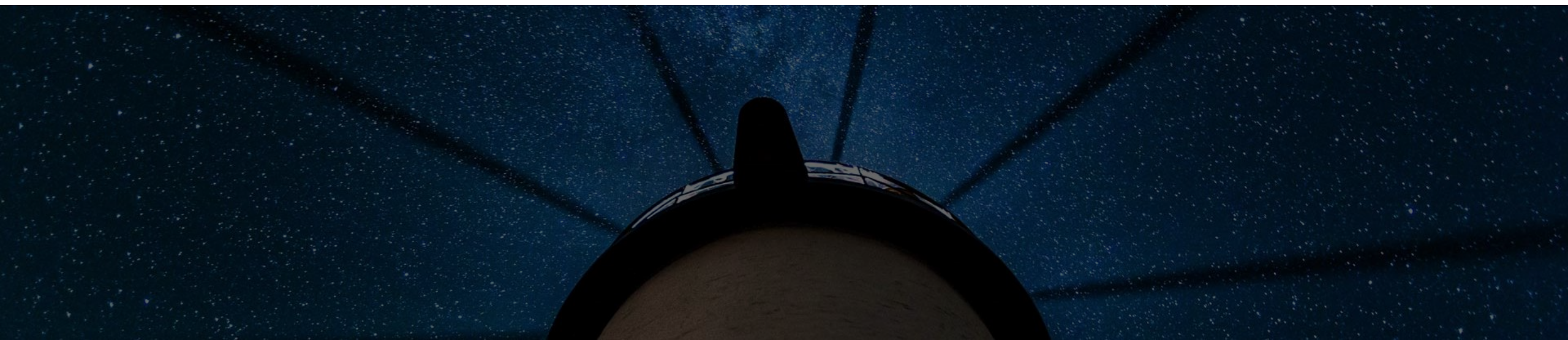
03

应用和落地

NI MEIYOU ZHUYI DAO BA

01 静态网站安全性

SECURITY OF STATIC WEBSITES



相关信息

- 在金砖会晤和十九大期间，我们整个保障措施可以看许卓斌月初在安全工作组会议分享的《重点服务保障期间的安全实践》
<http://sec2017.eventdove.com/>
- 静态网页安全防护可看《中国教育网络》2017年11月 P55页。电子版暂缺。
- DNS 安全：<https://dog.xmu.edu.cn/images/paper/dns-query-log-analysis-system-based-on-open-source-software.pdf>

网络安全、漏洞、攻防、应急响应

厦门大学

重大活动时期网站安全防护手册

厦门日报出品

在大型活动保障期间，为了保障网站的安全性，减少泄密压力，信息部门会选中或定制网站安全防护系统，系统会部署于最顶层的服务器中，门户网站中为学校的“第一入口”，往往也是黑客们攻击目标，因此我们门户网站的安全防护成为大型活动保障的重要工作。

将整个网站静态化，可以减少服务器动态脚本语言带来的安全风险。在入侵过程中，XSS、CSRF 等安全威胁，静态化允许不能完整展现所有页面，通过搜索引擎的静态化页面访问，可以防止黑客通过搜索引擎、爬虫等手段对网站进行攻击。静态化网站不能完整展现所有页面，通过搜索引擎的静态化页面访问，可以防止黑客通过搜索引擎、爬虫等手段对网站进行攻击。静态化网站不能完整展现所有页面，通过搜索引擎的静态化页面访问，可以防止黑客通过搜索引擎、爬虫等手段对网站进行攻击。

一个网站页面，从服务器到浏览器之间会经过许多环节，中间任何一个环节都有可能被攻击，防护得当会大大降低安全风险。一般来说，只有静态化网站，黑客才不能随意篡改网站内容。静态化网站的安全防护存在复杂性和风险的可能，然而基于服务器防护的局限，通过多

层防御的安全防护是动态的，即使第一个环节没有攻击成功，也可以通过其他环节的防护。

静态化处理和安全性检查

如果网站本身是由网站程序生成的静态化页面，那么可以提前将网站生成的静态化页面部署到服务器上，通过静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。

1. 如果网站本身是由网站程序生成的静态化页面，那么可以提前将网站生成的静态化页面部署到服务器上，通过静态化技术将网站的动态内容转换为静态文件。2. 静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。3. 静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。

应当将静态化网站部署到服务器上，通过静态化技术将网站的动态内容转换为静态文件。

静态网站防护

一旦做好了静态化网站防护，那么就可以通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

色，使用 nginx 进行反向代理，安装 CheckSV 定期检测病毒，安装 chkrootkit 或 RootKit Hunter 定期查杀木马。

对于可能的 DNS 攻击和病毒检测，应当实时性地将静态化网站部署到服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

平台部署的病毒检测程序，通过定期检测下载所有内容进行检测，通过静态化技术将网站的动态内容转换为静态文件。

6. 灾备人员：静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

大活动保障期间应当定期检测，灾备人员 24 小时值班，对网站进行实时监控和应急响应，应做好内容备份和网站数据的备份。

4. 应急响应：静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

5. 网站安全防护措施：静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

运维安全的其他注意事项

1. 使用 Ansible 等自动化配置工具：静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

网站安全防护措施

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

网站安全防护措施

静态化技术将网站的动态内容转换为静态文件，这些静态文件可以部署在服务器上，通过静态化技术将网站的动态内容转换为静态文件。1. 网络层防护：通过静态化技术将网站的动态内容转换为静态文件。2. 应用层防护：通过静态化技术将网站的动态内容转换为静态文件。3. 数据层防护：通过静态化技术将网站的动态内容转换为静态文件。

（内容来源于网络及作者个人观点）

姜开达：网络安全的特点是攻防高度不对称，黑客可以一点突破引发目标全线奔溃。作为防护的一方，全局体系任何一处都不能存在短板。

用户端浏览器 -> hosts文件 -> DNS -> 浏览器缓存 -> Safe Browsing -> 证书检查 -> 网关/无线/路由器 -> 代理 -> CDN -> 防火墙 -> IPS/IDS -> WAF -> 负载均衡 -> 虚拟化宿主机 -> 虚拟机 -> 操作系统 -> HTTP服务器 -> HTTP服务器 WAF -> URL重写 -> 文件载入 -> 脚本解析 -> URL路由分发 -> Cache -> 数据库 -> 渲染结果 -> 压缩 -> HTTPS加包 -> 浏览器接收 -> 浏览器插件 -> JS渲染



《Westworld》 Teddy

简单和冗余的平衡

漏洞扫描

/FW/WAF/IPS/IDS/DDOS/Lynis/OpenScap/nmap/ClamAV/chkrootkit/RootKit

Hunter/Fail2ban/mod_evasive/mod_reqtimeout/mod_qos/OSSEC/mod_security

纵深防御

奥卡姆剃刀原则：若无必要，勿增实体

KISS (Keep It Simple, Stupid)

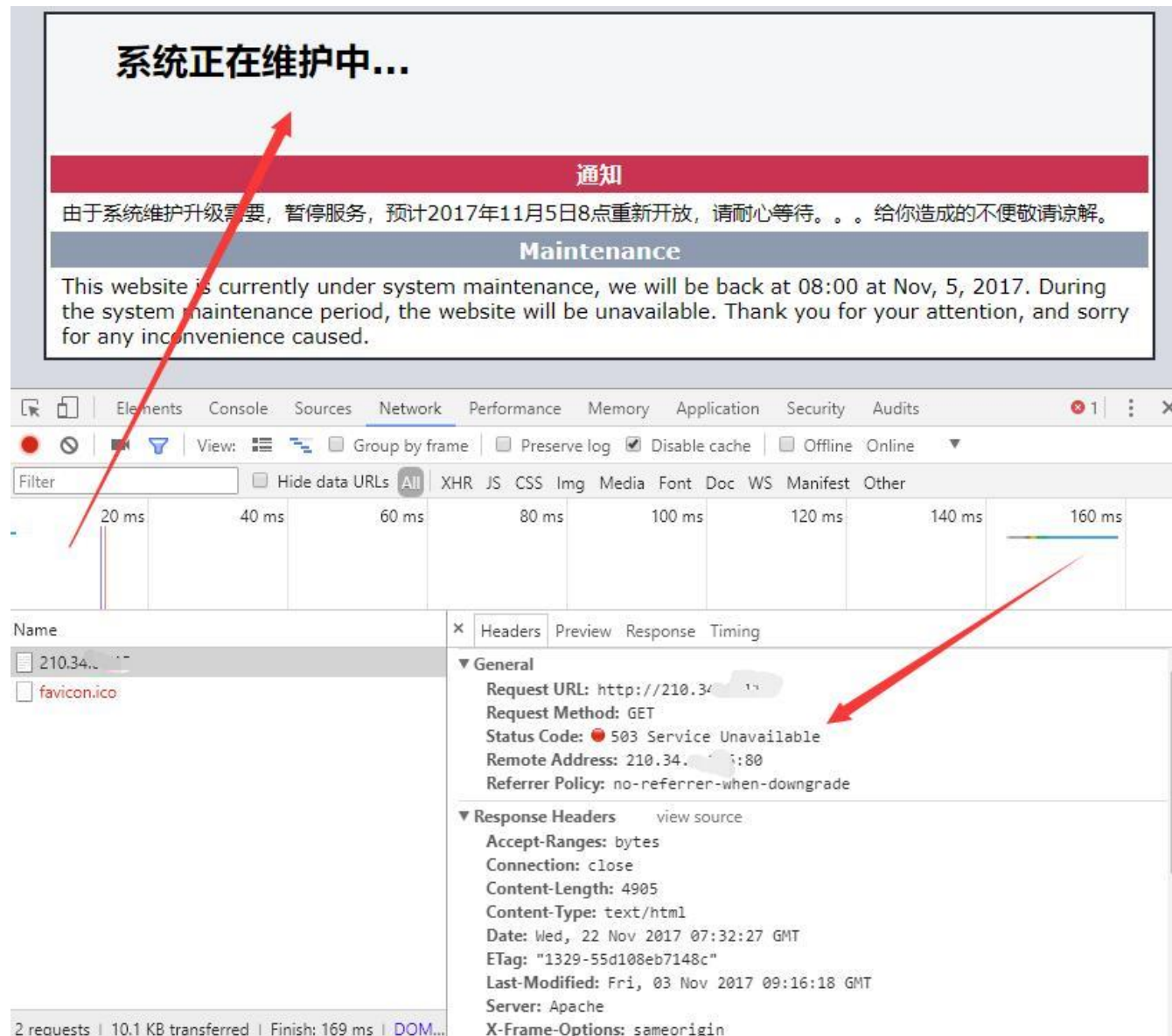


怼用户：你的问题，不是我的问题

云DNS投毒、CDN投毒、客户端本身ARP攻击、客户端接入商随意插入广告甚至被人截图后PS修改等。如发现应当及时下线，并执行检查，如果确定非自身因素则应当及时上线，并在页面上显著位置公布以消除不良影响。

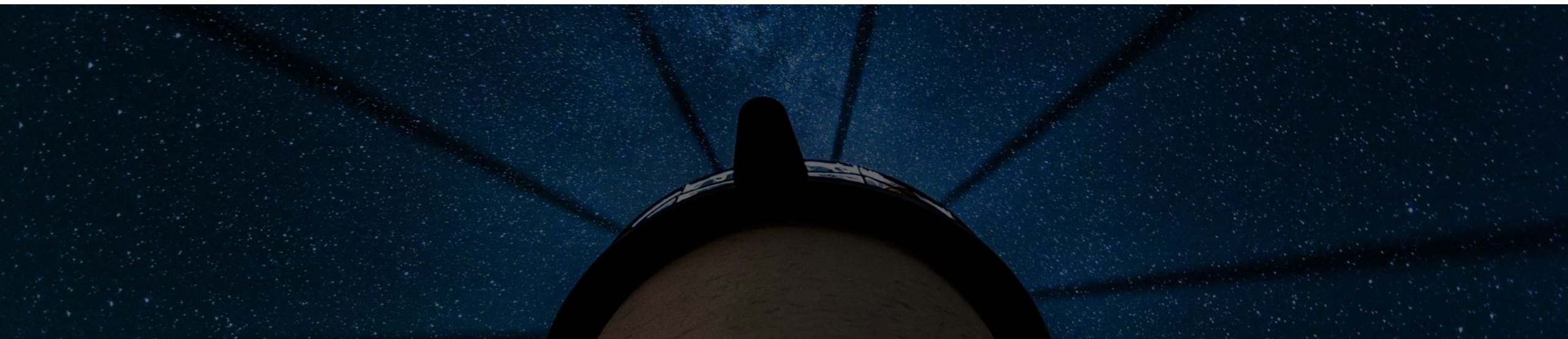
通知页面的技巧

- 对于一些存在安全隐患而被暂时性下线的网站，可使用应用交付设备或者把DNS导入到一个特定的通知页面，以减少突然关闭对网站管理员和浏览者带来的不便。
- 为避免临时性替换网站页面内容导致搜索引擎删除原有网站信息，通知页面应当返回503 HTTP状态码，也可根据恢复时间指定Retry-After返回值。



02 DevOps介绍

INTRODUCTION



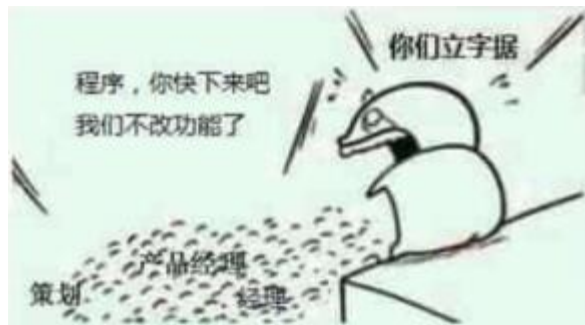
DevOps介绍

DevOps (Development和Operations的组合词) 是一种重视 “软件开发人员 (Dev) ” 和 “IT运维技术人员 (Ops) ” 之间沟通合作的文化、运动或惯例。透过自动化 “软件交付” 和 “架构变更” 的流程, 来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。 wikipedia.org

DevOps是一种方法论, 其中包含一系列基本原则和实践。

<http://www.infoq.com/cn/articles/detail-analysis-of-devops>

宽以待己，严以律人的开发



David is a DEVELOper !



David wants to
maximize
change

**Wall of
Confusion**

Peter is an OPERator !



Peter wants to
optimize
stability

话说天下大势，分久必合，合久必分

开发和运营之间的矛盾：

开发不希望产品经理频繁变更需求，运营不希望开发随意变更软件运行环境。

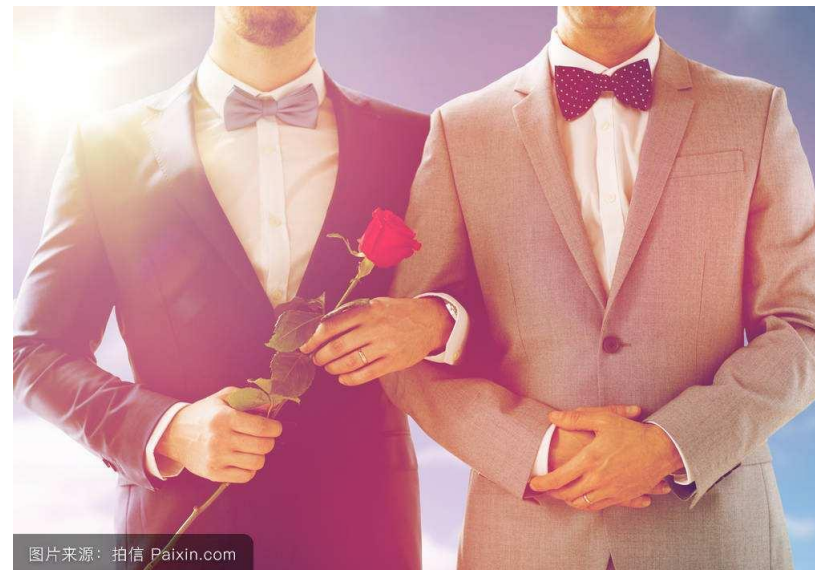
“在我的电脑上运行得好好的”

开发只关注实现功能，性能让运维处理。

分层，关注点分离

Dev和Ops结合，Dev不能只搞开发，也要去蹚运维的浑水。运维要把触角伸到开发。

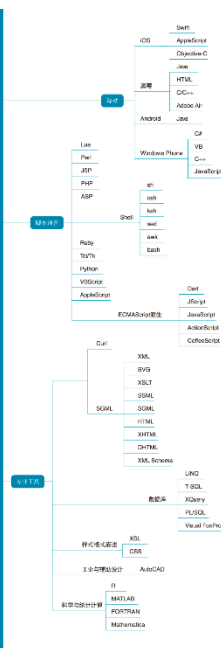
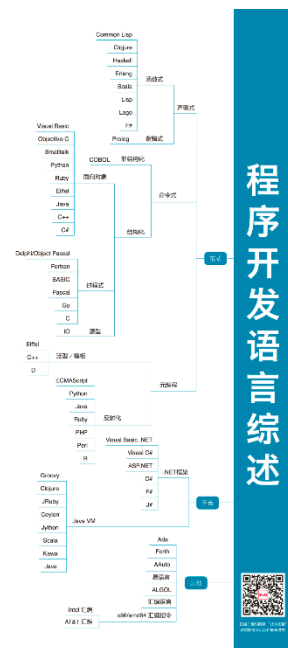
新名词：DevSecOps



带来新的问题

技能点要求不同：前端工程师、后端工程师、iOS工程师、Android工程师、架构师、DBA、网络工程师、基础架构工程师、发布经理。

成本不同。



StuQ IT职业技能图谱 V0.1.2

Geekbang

<https://github.com/TeamStuQ/skill-map>

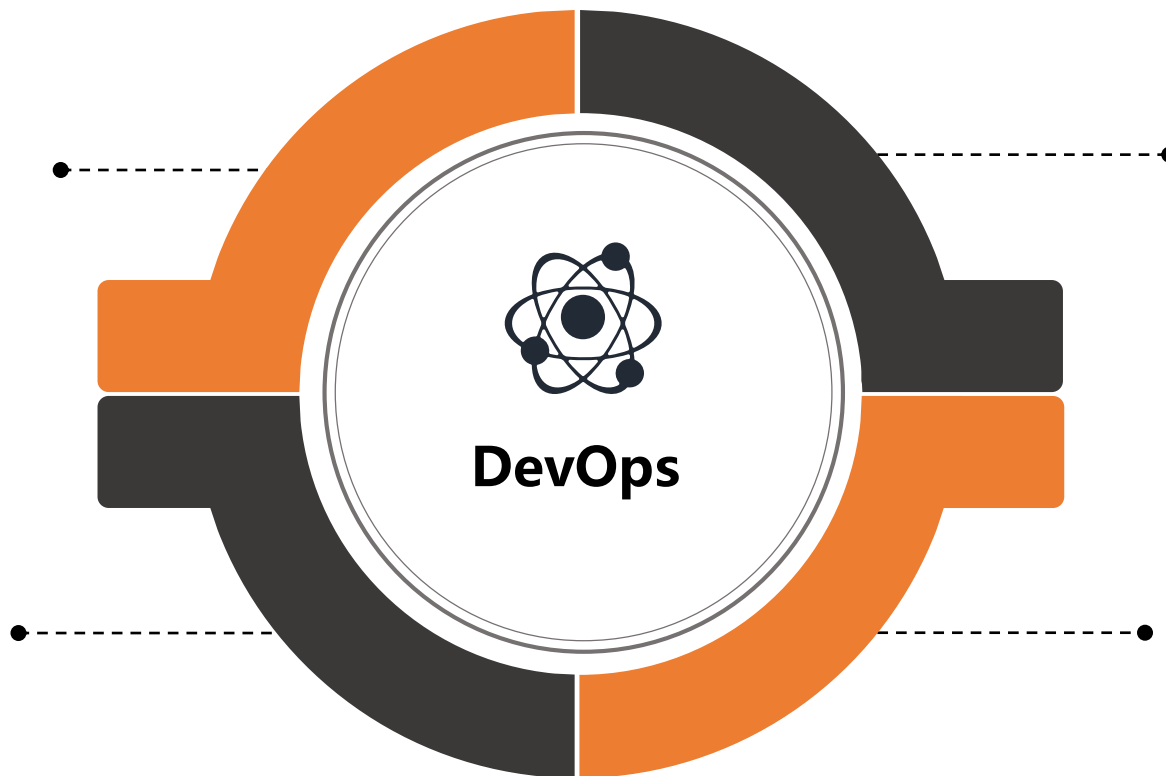


基础架构代码化

- 解放效率，可复制。
- 可重复运行，幂等 1^n 。
- 减少人为出错，执行人员可替换。
- 代码就是文档，方便审计。
- 版本控制，变更记录。

反馈快

走太快容易扯着X，小步快跑
每天部署几百次
编译测试要快



不可变服务器

- 改变部署的方法，原先替换一个文件，现在可以替换整个服务器。
- 打补丁更安全。
- 蓝绿部署。
- 金丝雀发布。

部署流水线

管道，链式调用，部署流水线，Gulp或者Grunt

管道示例：

```
cat dog.txt | sort | uniq -c | sort -n -r -k 1 -t ' ' | awk -F '/' '{print $2}' | head -5
```

JavaScript链接调用示例：

```
d3.select("#wordcloud_topN_by_post")
  .append("svg")
  .style("width", "100%")
  .attr("preserveAspectRatio", "xMidYMin meet")
  .append("g")
  .attr("transform", "translate(100, 100)")
  .selectAll("text")
  .data(words).enter().append("text")
  .style("font-size", function(d) { return d.size + "px"; })
  .style("font-family", "Impact")
  .style("fill", function(d, i) { return color(i); })
  .attr("text-anchor", "middle")
  .text(function(d) { return d.text; })
```

部署流水线：开发 -> 单元测试 -> 提交Git -> Pull代码 -> 编译 -> 测试 -> JavaScript混淆整合最小化版本化 -> 图片优化整合 -> 打包编译Docker -> 自动化集成测试 -> 正式环境配置 -> 数据库Migrate -> 蓝绿发布 -> 正式发布 -> 监控

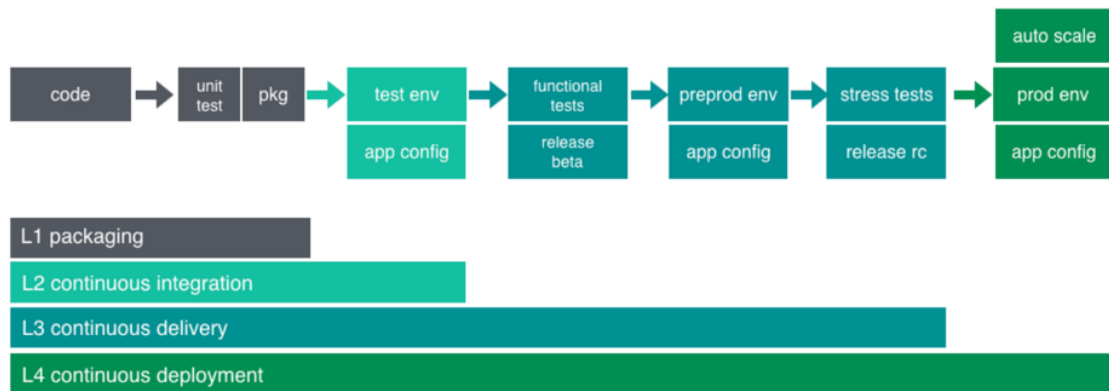
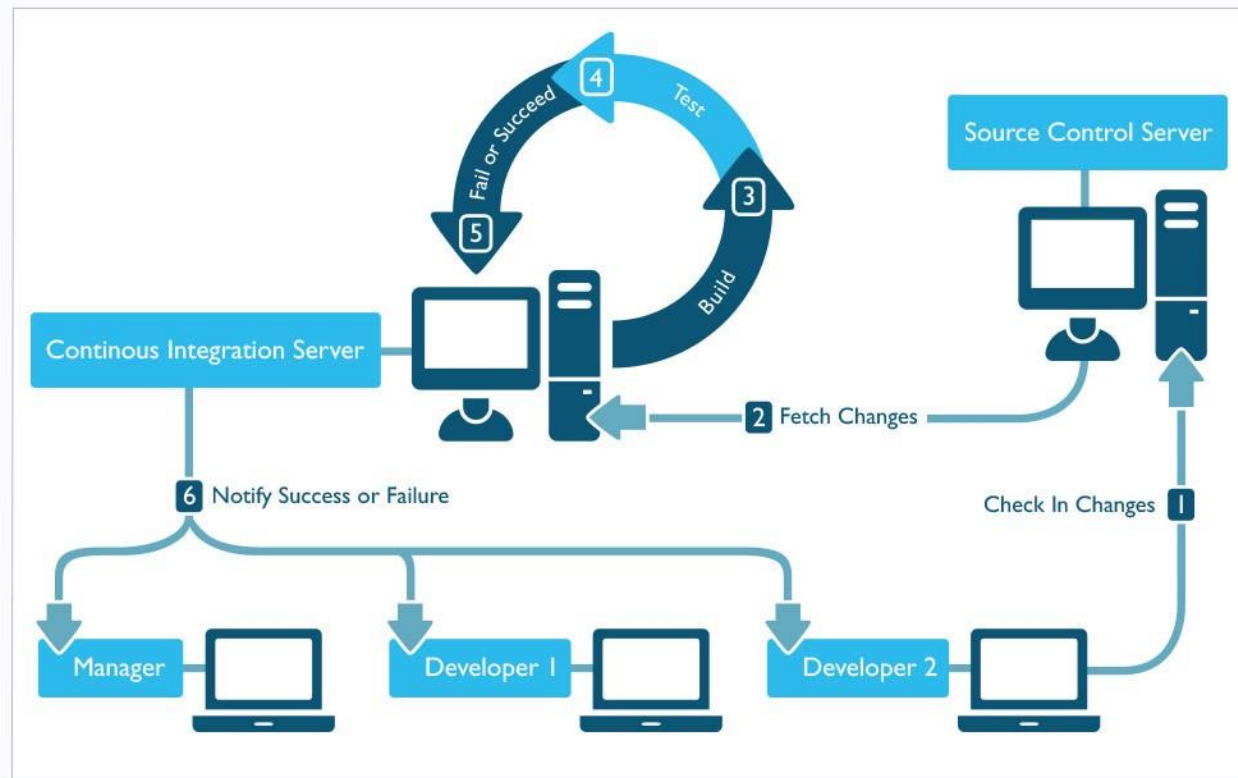


Figure: Continuous deployment maturity model

<https://dzone.com>



<http://insights.sei.cmu.edu>

Why Ansible

Ansible, 主控模式, 依赖小, 通过SSH, 基于Python, YAML语法。顺序执行。

Puppet, Pull模式。Ruby, DSL语法。执行顺序必须显式写出依赖。

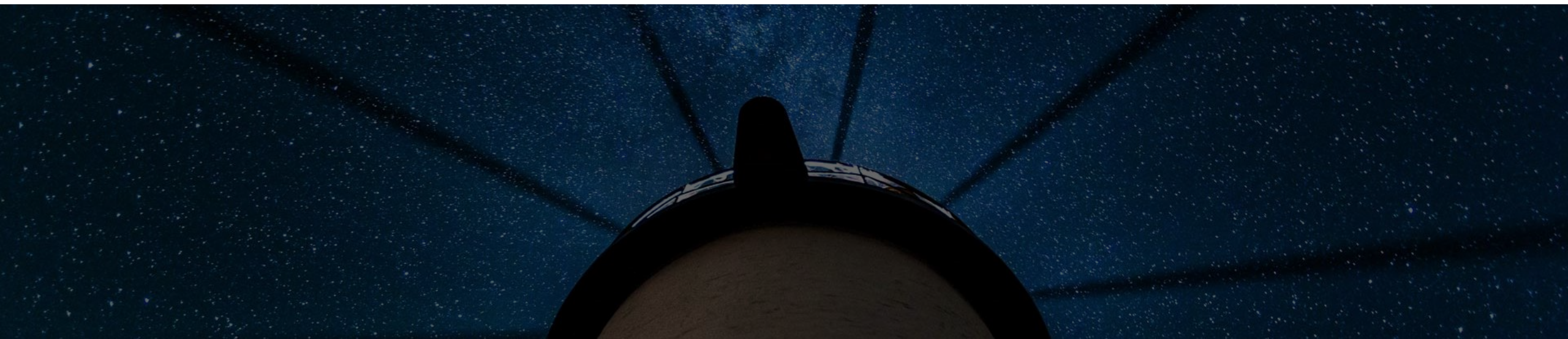
Chef

SaltStack

CFEngine

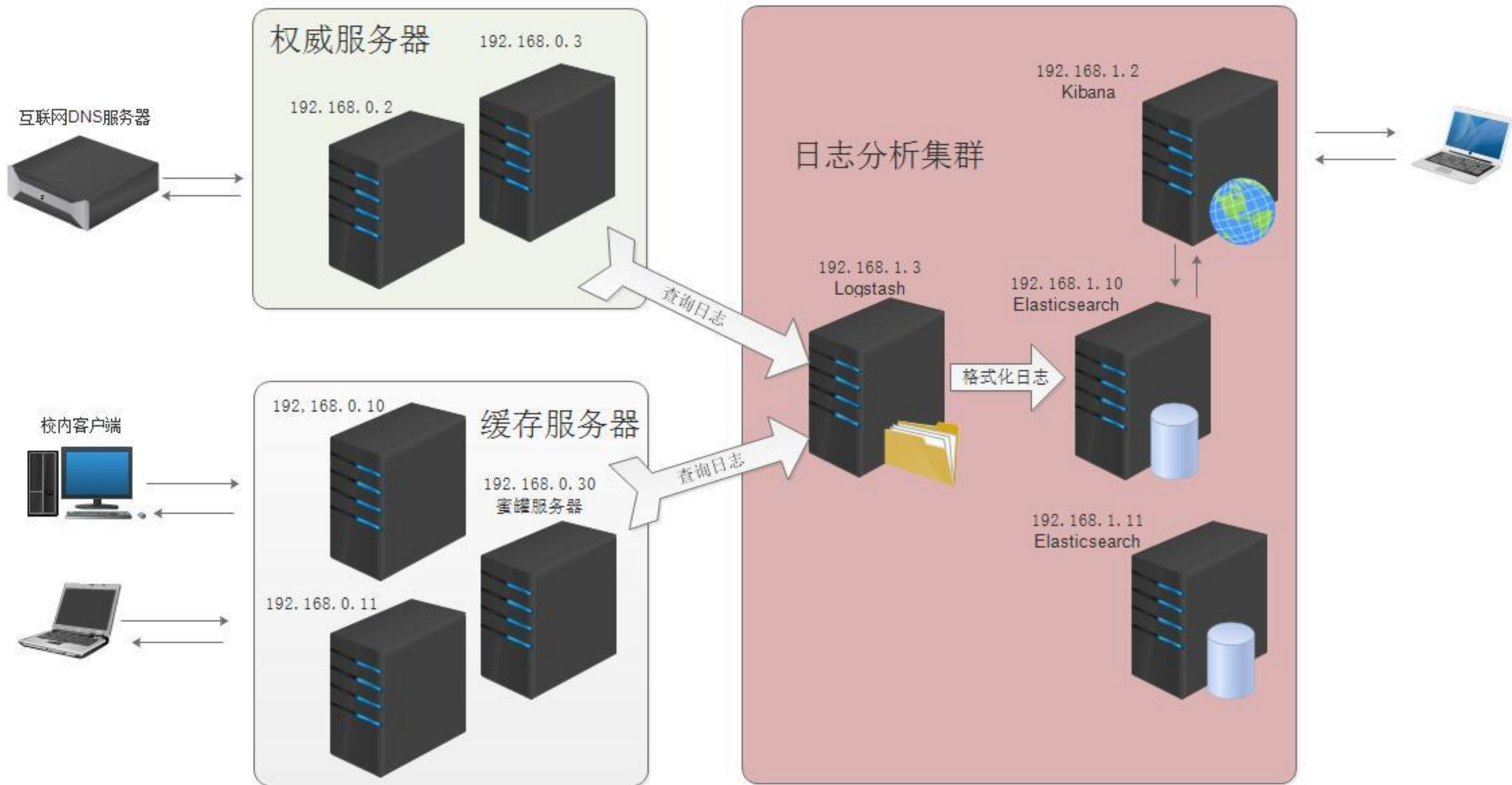
03 应用和落地

PRACTICE



厦门大学主页安全保障中应用

- 基础架构代码化
- 静态网站生成自动化：wget下来，Python脚本更改链接地址，PhantomJS截图
- DNS自动化：更改条目后，跑一个命令reload.sh，自动比对所有DNS条目是否有备案，没有备案劫持到通知页面，域名指向校外IP直接忽略。
Master/Master 架构复制，更新缓存服务器的.xmu.edu.cn域，dig测试view和4台服务器解析是否正确。
- 巡检自动化：Ansible获取Facts，比对。



ansible-playbook site.yml -i hosts.production

- ▼ dns
 - ▶ group_vars
 - ▶ keys
 - ▶ other
- ▼ roles
 - ▶ common
 - ▶ elasticsearch
 - ▶ filebeat
 - ▶ kibana
 - ▶ logstash
 - ▼ master
 - ▶ files
 - ▶ handlers
 - ▶ tasks
 - ▶ templates
 - ▶ resolver
- .gitignore
- hosts.real
- hosts.test
- README.md
- site.retry
- site.yml

```
1 [master-server]
2 210.34.0.1 is_controller=true
3 210.34.0.2 is_controller=false
4
5 [resolver-server]
6 210.34.0.1
7 210.34.0.2
8
9 [elasticsearch-server]
10 172.27.64.1 jvm_memory_minmax=26
11
12 [logstash-server]
13 172.27.77.1
14
15 [kibana-server]
16 172.27.77.1
```

hosts.production定义IP和服务名

```
1 ---
2 - name: master
3   hosts: master-server
4   roles:
5     - common
6     - master
7     - { role: filebeat, document_type: 'nslog' }
8
9 - name: resolver
10  hosts: resolver-server
11  roles:
12    - common
13    - resolver
14    - { role: filebeat, document_type: 'dnslog' }
15  tags:
16    - debug
17
18 - hosts: elasticsearch-server
19  roles:
20    - common
21    - elasticsearch
22
23 - hosts: logstash-server
24  roles:
25    - common
26    - logstash
27
28 - hosts: kibana-server
29  roles:
30    - common
31    - kibana
```

site.yml定义服务器的角色

- ▼ resolver
 - ▼ files
 - db.rzp.xmu.evil
 - named.conf.options
 - rndc.key
 - ▼ handlers
 - main.yml
 - ▼ tasks
 - main.yml
 - ▼ templates
 - named.conf.local.j2
 - user.rules.j2
 - user6.rules.j2

- task/main.yml定义角色的具体操作内容
- 我需要安装什么软件
- 我的软件如何配置（修改配置文件）

```
- name: Update all packages to the latest version
  apt:
```

```
    update_cache: yes
    cache_valid_time: 864000 # 86400 1 day
    upgrade: dist
    autoclean: yes
    autoremove: yes
    purge: yes
    become: true
```

```
- name: install apache2
```

```
  package:
    pkg: apache2
    state: installed
    become: true
```

```
- name: install php7.0-xml
```

```
  package:
    pkg: php7.0-xml
    state: installed
    become: true
    notify: restart apache2
```

```
- file: path=/var/www/cache/ state=directory owner=www-data group=www-data
  become: true
```

```
- name: copy www.conf
```

```
  template: src=www.conf.j2 dest=/etc/apache2/sites-available/www.conf owner=root group=root mode="a+r"
  vars:
    conf_type: http
    become: true
    notify: restart apache2
```

```
- apache2_module:
```

```
  state: absent
  name: "{{ item }}"
  force: true
  become: true
  with_items:
    - autoindex
    - alias
  notify: restart apache2
```

```
- apache2_module:
```

```
  state: present
  name: "{{ item }}"
  become: true
  with_items:
    - ssl
    - rewrite
    - headers
  notify: restart apache2
```

talk is cheap
show me the
CODE

```
- ufw:
  state: reset
  become: true

- ufw:
  state: enabled
  policy: deny
  become: true

- ufw:
  rule: allow
  port: 22
  with_items:
    - {{host}}

- ufw:
  rule: allow
  port: 5044
  with_items:
    - {{host}}
```

```
- ufw:
  state: enabled
  policy: deny
  become: true
  tags: testufw

- name: copy user.rules
  template: src=user.rules.j2 dest=/etc/ufw/user.rules owner=root group=root
  become: true
  notify: restart ufw
```

user.rules.j2

```
{% for host in groups['master-server'] %}
  {% if hostvars[host].is_controller == 'true' %}
### tuple ### allow tcp 22 0.0.0.0/0 any {{host}} in
-A ufw-user-input -p tcp --dport 22 -s {{host}} -j ACCEPT
  {% endif %}
{% endfor %}
```

```
{% for host in groups['master-server'] + groups['resolver-server'] %}
### tuple ### allow tcp 5044 0.0.0.0/0 any {{host}} in
-A ufw-user-input -p tcp --dport 5044 -s {{host}} -j ACCEPT
{% endfor %}
```

```
TASK [kibana : start kibana] *****
ok: [172.27.1.7]

TASK [kibana : lineinfile] *****
ok: [172.27.1.7]

TASK [kibana : lineinfile] *****
ok: [172.27.1.7]

PLAY RECAP *****
172.27.1.7 : ok=31 changed=0 unreachable=0 failed=0
172.27.1.8 : ok=17 changed=0 unreachable=0 failed=0
172.27.1.9 : ok=18 changed=0 unreachable=0 failed=0
210.34.1.5 : ok=28 changed=0 unreachable=0 failed=0
210.34.1.6 : ok=36 changed=0 unreachable=0 failed=0
210.34.1.7 : ok=35 changed=0 unreachable=0 failed=0
210.34.1.8 : ok=28 changed=0 unreachable=0 failed=0
```

conf.d 目录

apache: /etc/apache2/conf-enabled

crontab: /etc/cron.daily

included local_setting.conf

```
- lineinfile:
  path: /etc/apache2/conf-enabled/security.conf
  regexp: '^ServerTokens'
  line: 'ServerTokens Prod'
  become: true
  notify: restart apache2

- lineinfile:
  path: /etc/apache2/conf-enabled/security.conf
  regexp: '^ServerSignature'
  line: 'ServerSignature Off'
  become: true
  notify: restart apache2

- lineinfile:
  path: /etc/apache2/conf-enabled/security.conf
  regexp: '#?Header set X-Frame-Options: "sameorigin"'
  line: 'Header set X-Frame-Options: "sameorigin"'
  become: true
  notify: restart apache2

- blockinfile:
  path: /etc/elasticsearch/elasticsearch.yml
  block: |
    indices.breaker fielddata.limit: 9gb
    http.cors.enabled: true
    http.cors.allow-origin: "*"
  become: true
  notify: restart Elasticsearch
```


A top-down view of a desk. In the upper left, a white keyboard is partially visible. To its right is a row of colorful books standing upright. Below the keyboard is a magazine with a blue cover featuring an iPod and the text 'iPod и iPod mini'. In the lower left, there's a spiral-bound notebook with a pencil and an eraser. In the center, a smartphone is placed on top of an open book. The book's pages show various logos like 'Quark', '@', 'conEdison', and 'University Science Park'.

谢谢

厦门大学 郑海山
<https://dog.xmu.edu.cn>

标注

字体使用

英文 微软雅黑

中文 微软雅黑

行距

正文 1.3

背景图片出处

OfficePLUS Pexels

模板作者

Healthyyang2

OfficePLUS