



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

基于SaltStack的基础设施即代码

赵班长 中国SaltStack用户组发起人

自我介绍



GOPS2018
Shenzhen

赵舜东

- 花名：“赵班长”，曾在武警某部负责指挥自动化的架构和运维工作，2008年退役后一直从事互联网运维工作，历任运维工程师、运维经理、运维架构师、运维总监。
 - 中国SaltStack用户组发起人（<http://www.saltstack.cn/>）
 - 运维社区创始人（<http://www.unixhot.com/>）
 - DevOps学院创始人（<http://www.devopsedu.com/>）
 - 著作：《SaltStack入门与实践》、《运维知识体系》、《缓存知识体系》
 - 中国首批Exin DevOps Master 认证讲师、GOPS金牌讲师



GOPS2018
Shenzhen

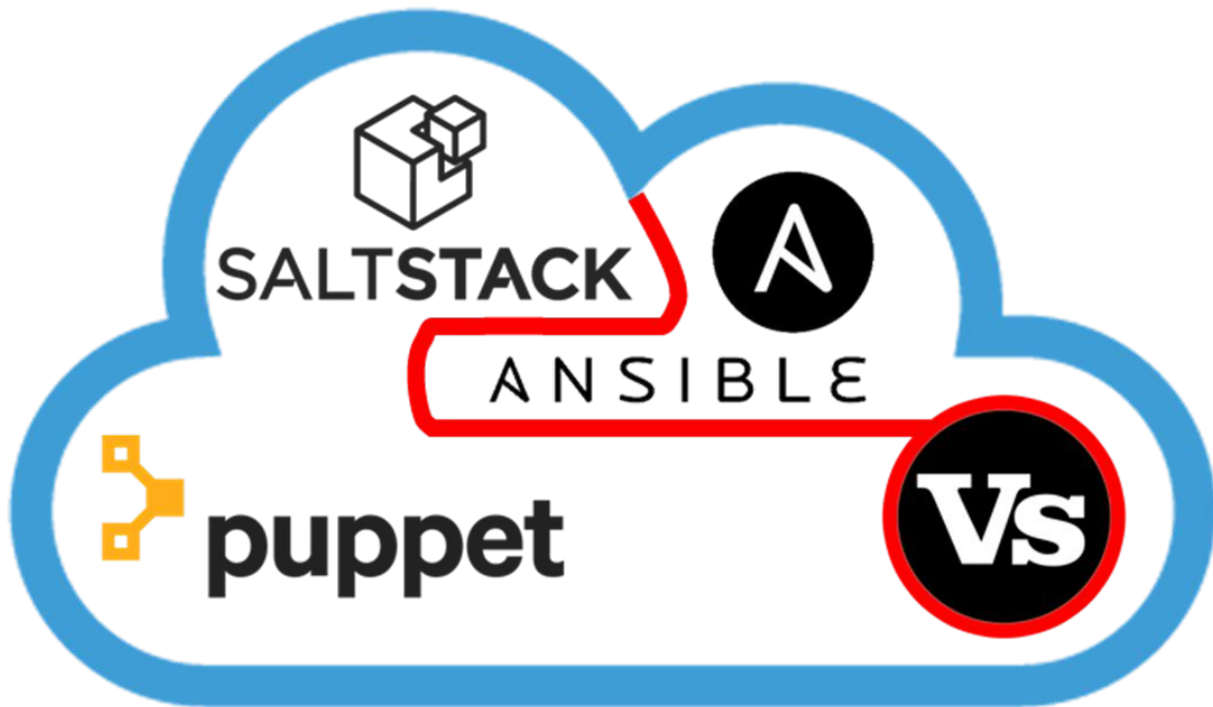
目录

- ➔ **1** 配置管理（状态管理）
- 2** 基于SaltStack的基础设施即代码
- 3** 以SaltStack为基础的自动化运维实践
- 4** Everything is Code
- 5** QA

你确定你是在使用配置管理？



GOPS2018
Shenzhen



状态管理的神话故事



GOPS2018
Shenzhen



谁是神笔？

SaltStack

谁是马良？

SaltStack使用者



GOPS2018
Shenzhen

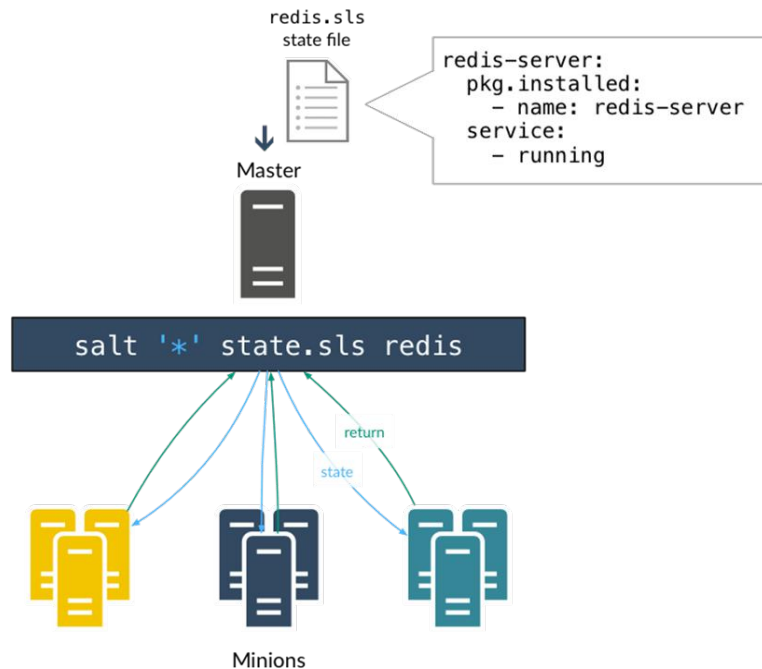
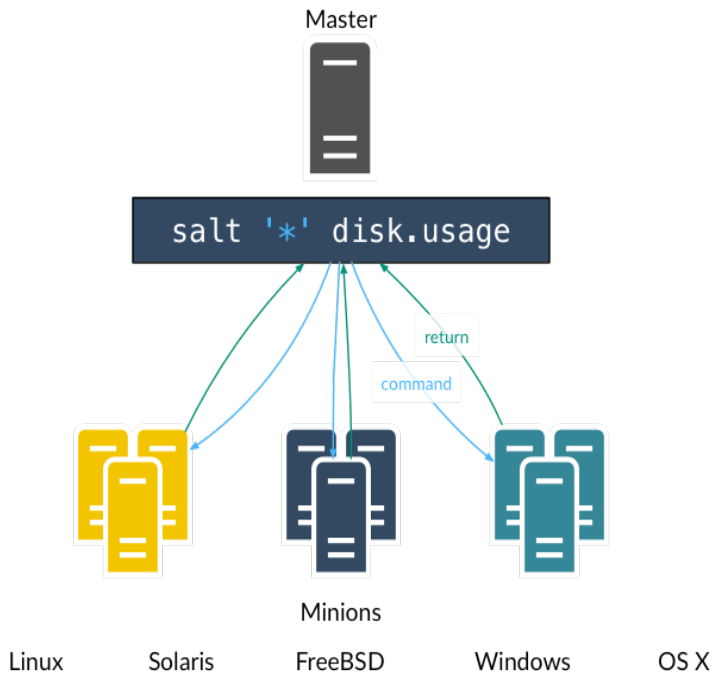
目录

- 1 配置管理（状态管理）
- ➔ 2 基于SaltStack的基础设施即代码
- 3 以SaltStack为基础的自动化运维实践
- 4 Everything is Code
- 5 QA

远程执行和配置管理



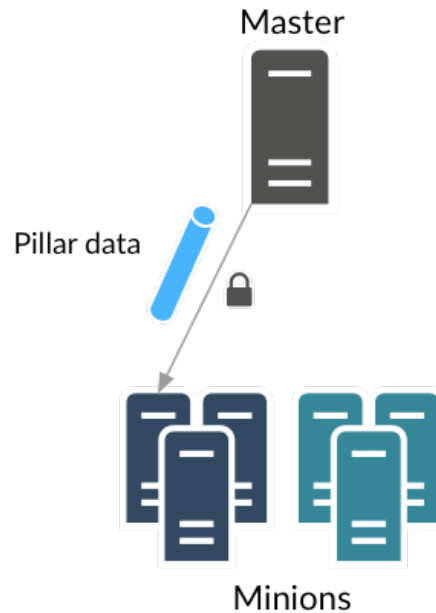
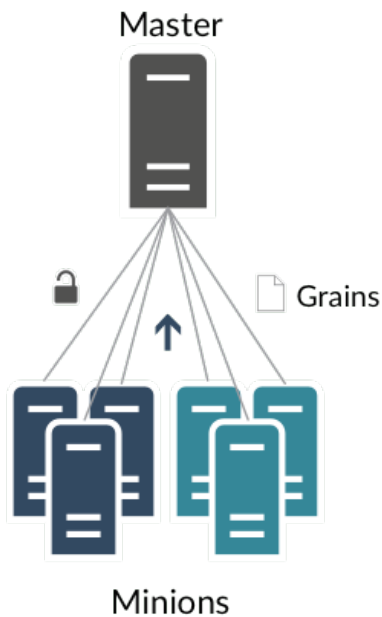
GOPS2018
Shenzhen



数据系统Grains Pillar

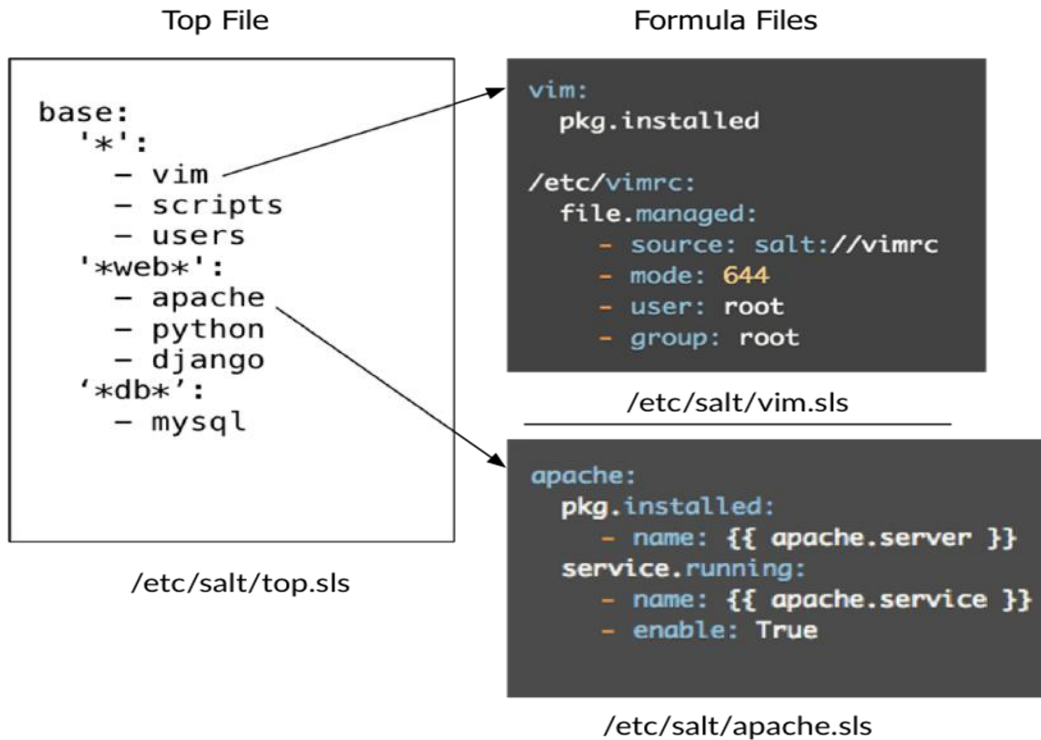


GOPS2018
Shenzhen





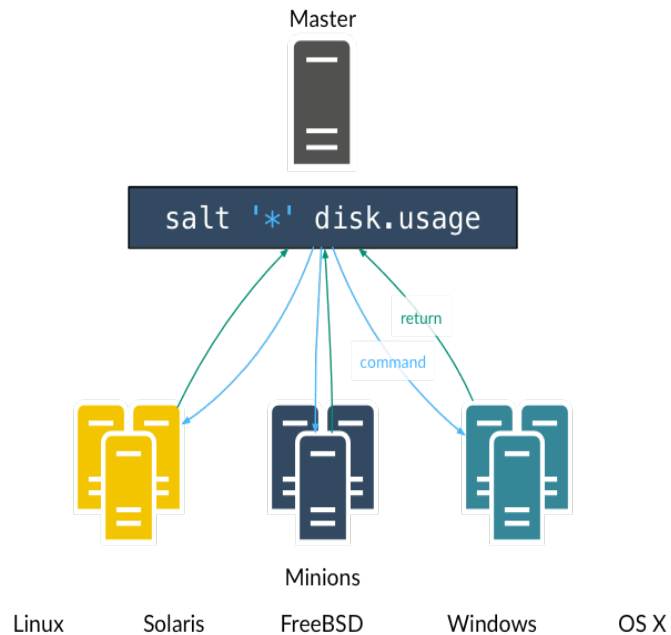
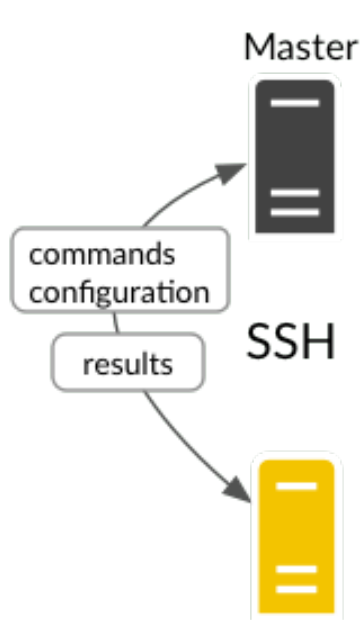
高级状态Top File



Salt SSH你值得拥有!



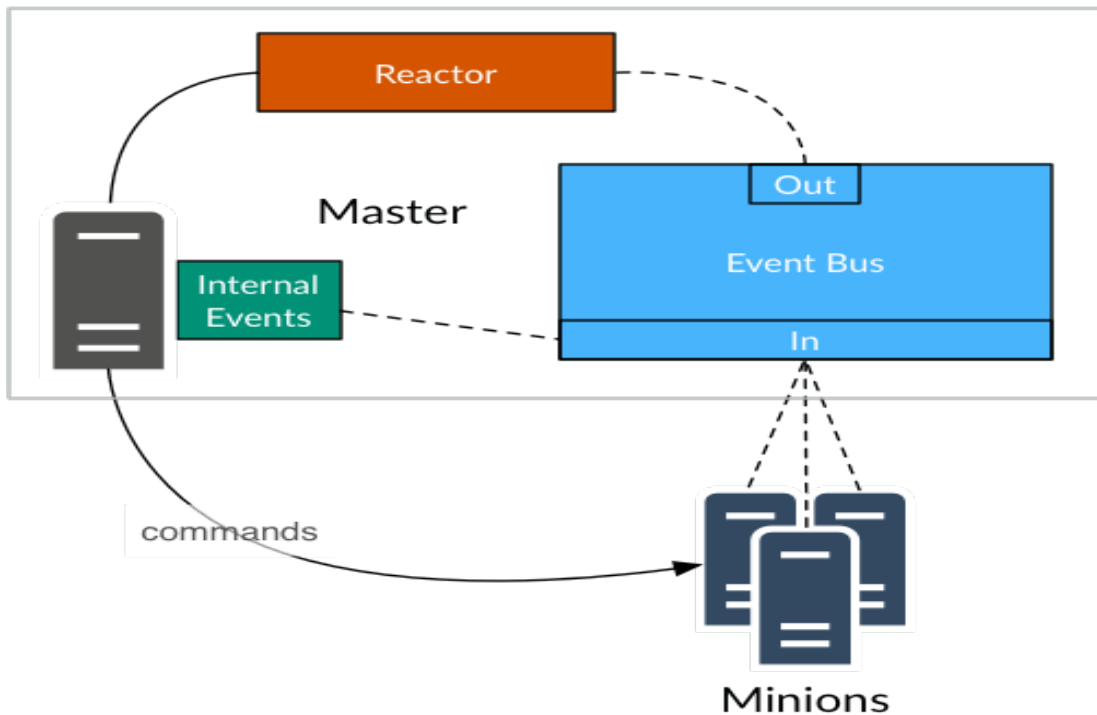
GOPS2018
Shenzhen



事件驱动基础设施



GOPS2018
Shenzhen





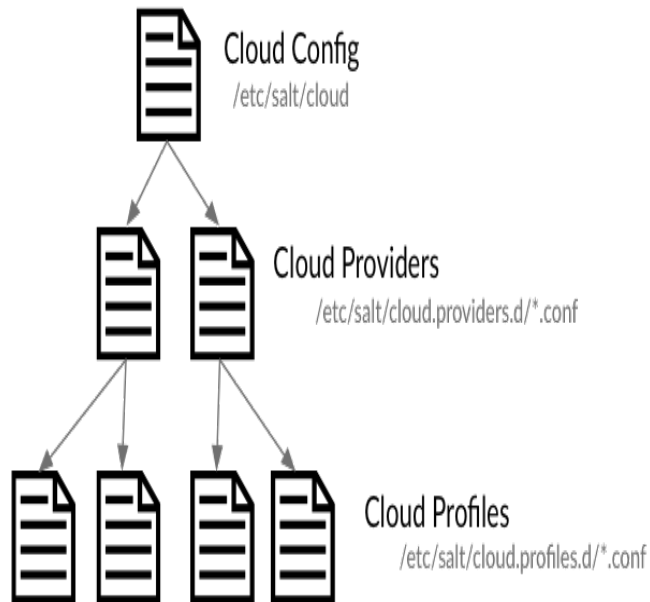
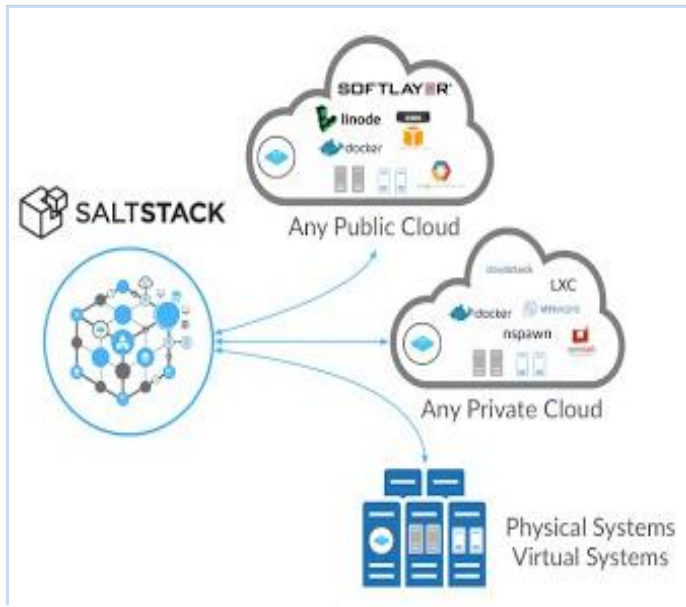
来个简单的Demo!

<https://github.com/unixhot/salt-example>





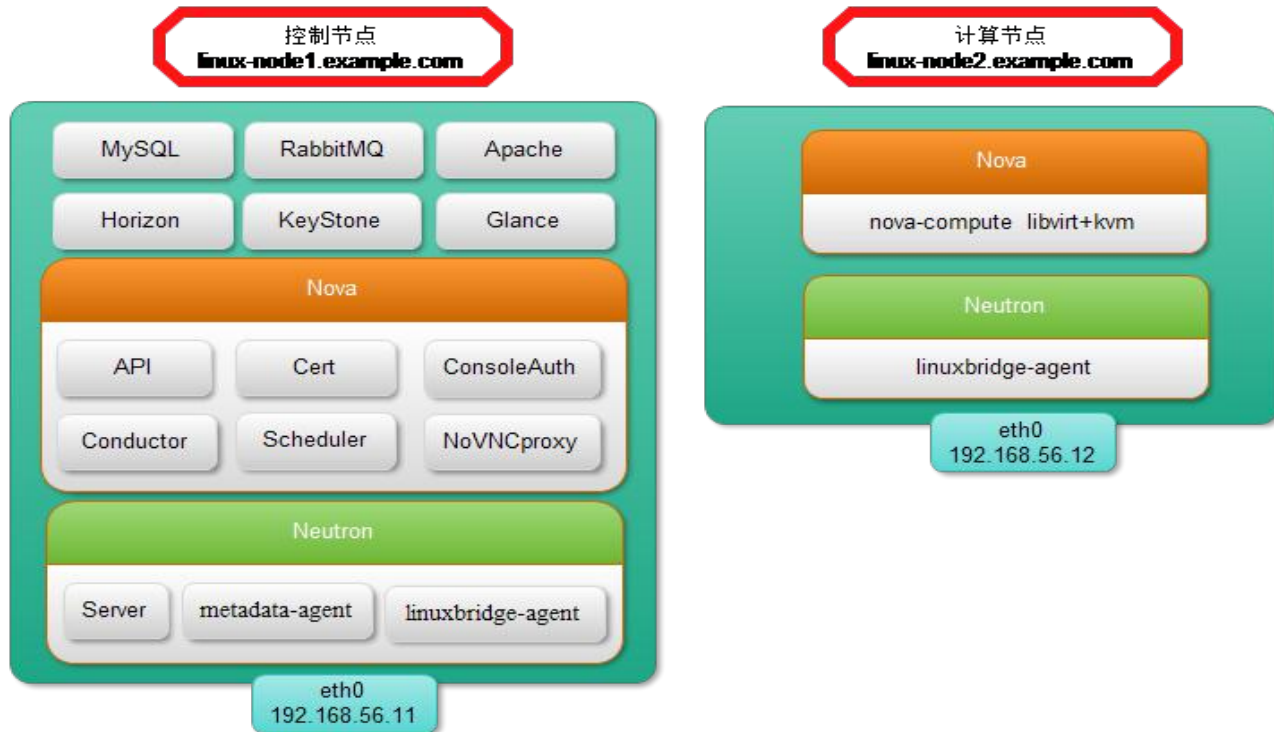
云计算时代还需要SaltStack吗？





再来个Demo！！

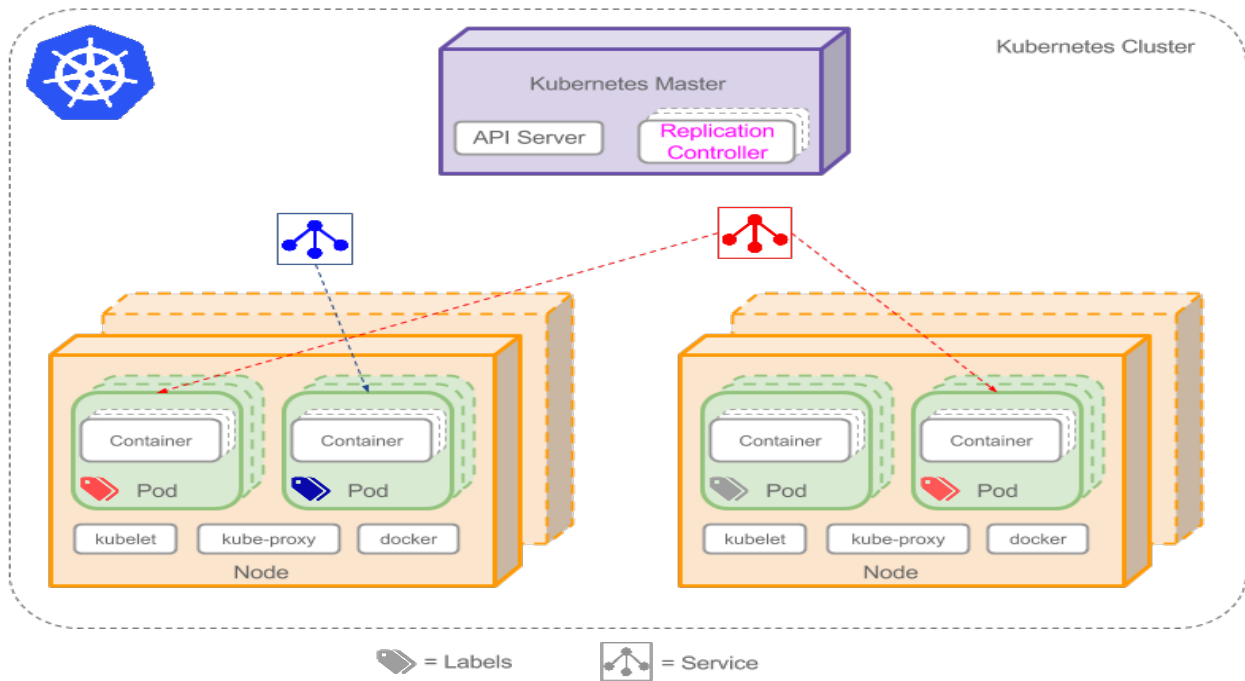
<https://github.com/unixhot/salt-openstack>





GOPS2018
Shenzhen

现在都容器化了，还扯什么SaltStack？





GOPS2018
Shenzhen

再再来个Demo!!!

<https://github.com/unixhot/salt-kubernetes>

SaltStack自动化部署Kubernetes v1.9.3版本（支持TLS 双向认证、RBAC 授权、Flannel网络、ETCD集群等）

版本明细：Release-v1.0

- 测试通过系统：CentOS 7.4
- salt-ssh: 2017.7.4
- Kubernetes：v1.9.3
- Etcd: v3.3.1
- Docker: 17.12.1-ce
- Flannel：v0.10.0
- CNI-Plugins：v0.7.0 建议部署节点：最少三个节点，请配置好主机名解析（必备）

架构介绍

1. 使用Salt Grains进行角色定义，增加灵活性。
2. 使用Salt Pillar进行配置项管理，保证安全性。
3. 使用Salt SSH执行状态，不需要安装Agent，保证通用性。
4. 使用Kubernetes当前稳定版本v1.9.3，保证稳定性。





GOPS2018
Shenzhen

目录

1 配置管理（状态管理）

2 基于SaltStack的基础设施即代码

➔ 3 以SaltStack为基础的自动化运维实践

4 Everything is Code

5 QA



GOPS2018
Shenzhen

结合CMDB完成基于角色的运行环境交付



- 1.使用ext_pillar做数据扩展
- 2.使用状态管理做交付

结合Zabbix做自动化监控套件



GOPS2018
Shenzhen

Zabbix监控套件组成:

1. 监控脚本
2. 自定义配置
3. 监控模板





GOPS2018
Shenzhen

目录

1 配置管理（状态管理）

2 基于SaltStack的基础设施即代码

3 以SaltStack为基础的自动化运维实践

➔ 4 Everything is Code

5 QA



基于SaltStack的基础设施即代码

```
1 include:
2   - modules.Keepalived.install
3
4 keepalived-server:
5   file.managed:
6     - name: /etc/keepalived/keepalived.conf
7     - source: salt://cluster/files/haproxy-outside-keepalived.conf.template
8     - mode: 644
9     - user: root
10    - group: root
11    - template: jinja
12    {% if grains['fqdn'] == 'linux-node1.example.com' %}
13    - ROUTEID: haproxy_ha
14    - STATEID: MASTER
15    - PRIORITYID: 150
16    {% elif grains['fqdn'] == 'linux-node2.example.com' %}
17    - ROUTEID: haproxy_ha
18    - STATEID: BACKUP
19    - PRIORITYID: 100
20  {% endif %}
21  service.running:
22    - name: keepalived
23    - enable: True
24    - watch:
25      - file: keepalived-server
```

19:40

开始加班接着写代码



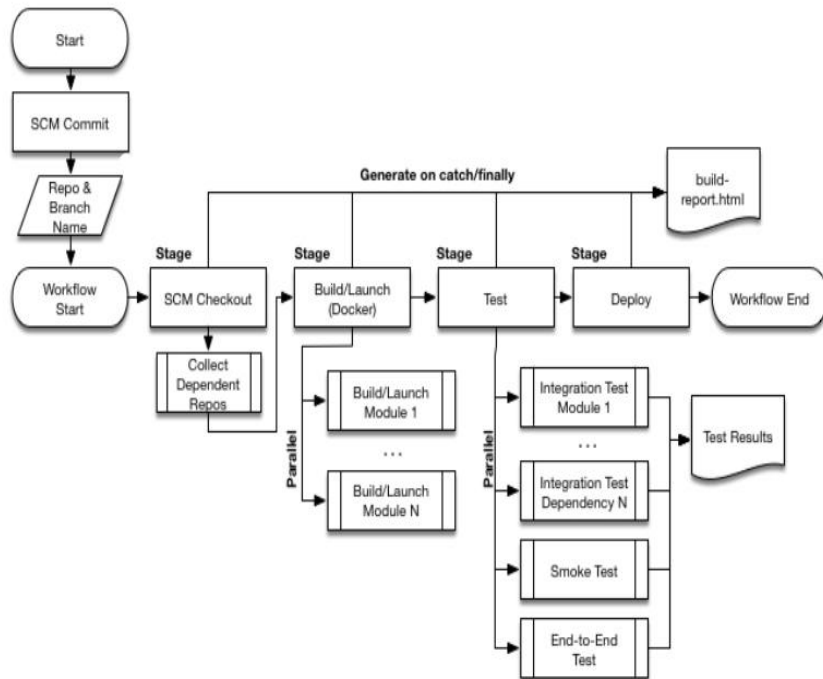


GOPS2018
Shenzhen

基于Jenkins的Pipeline as Code



Development



Production

基于Jenkins的Pipeline as Code



GOPS2018
Shenzhen

Open Blue Ocean

Full Stage View

Pipeline Syntax

Build History 构建历史

find

- #29 2017-8-4 下午3:47
- #28 2017-8-4 下午3:46
- #27 2017-8-4 上午11:35
Started by GitLab push by Administrator
- #26 2017-8-4 上午11:27
- #25 2017-8-4 上午11:25

RSS 全部 RSS 失败

Stage View

Average stage times:

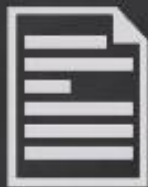
	Code Checkout	Build Code	Unit Test	Code Quality
	4s	7s	14s	16s
#29 Aug 04 15:47 No Changes	1s	5s	10s	11s
#28 Aug 04 15:46 No Changes	1s	5s	11s	11s
#27 Aug 04 11:35 3 commits	2s	7s	17s	21s
#26 Aug 04 11:27 No Changes	3s	8s	17s	19s
#25 Aug 04 11:25 No Changes	12s	9s	17s	19s

一切即代码

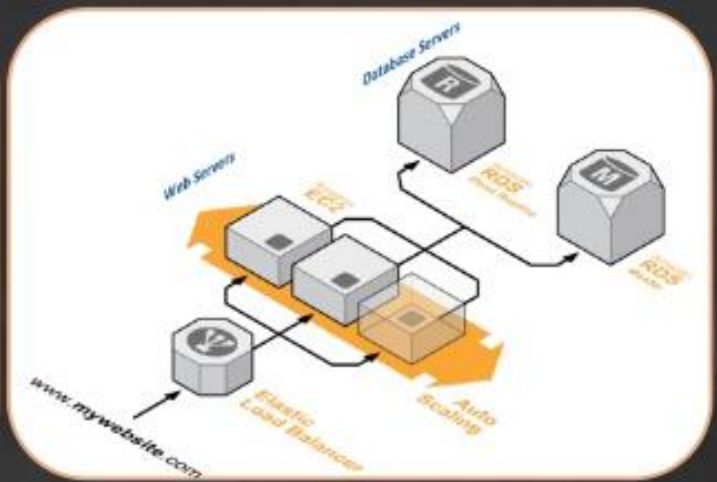


GOPS2018
Shenzhen

// Everything is code



CODE();



版本控制



GOPS2018
Shenzhen

目录

1 配置管理（状态管理）

2 基于SaltStack的基础设施即代码

3 以SaltStack为基础的自动化运维实践

4 Everything is Code

➔ 5 QA



GOPS2018
Shenzhen

Q & A



来！互相伤害啊





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
最新动态吗？

