saltstack 在运维平台上的应用

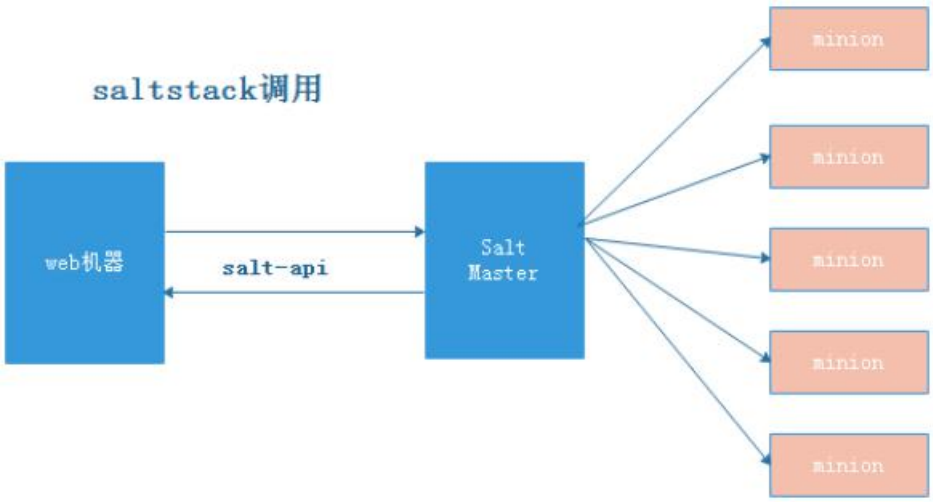
-----分享自競技世界_吴胜宝_2018 年 8 月 25 日

大概内容目录：

分为 key 管理、分组管理、执行部分和安全部分，以及 api 和运用

一、key 管理；主要是管理连接 master 的主机，主要功能是接受秘钥，拒绝秘钥和删除秘钥。主要使用 Salt-key list 扫描节点并入库，更新新增节点和失效节点。

先说下 salt 调用的主要过程：

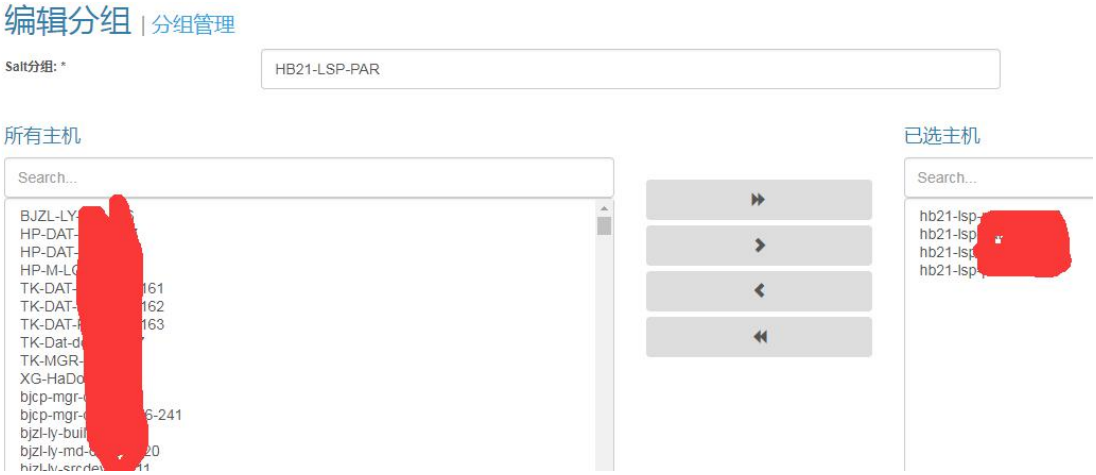


saltmaster 和 web 是分开的，master 有多台，web 机器和 master 之间通过 saltapi 来进行交互。先将指令下发给对应的 master，然后由 master 来往节点机器上下发执行。

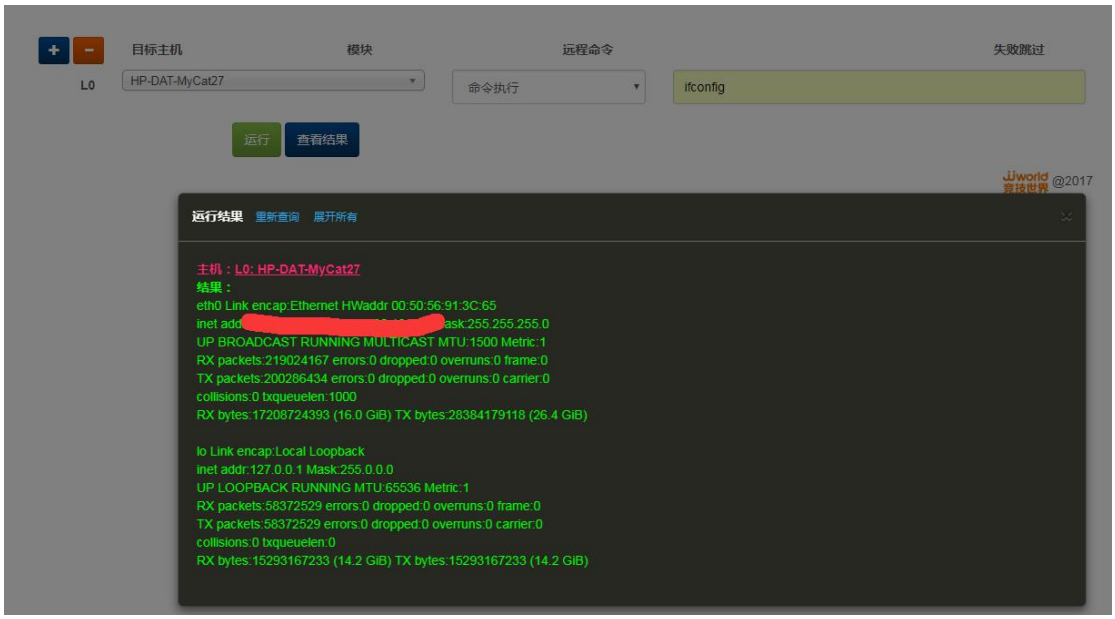
二、分组管理

节点数量很多，有时候就需要分组，在页面上进行分组，存入本地 mysql，然后将分组信息通过服务同步到 master 机器上，具体做法是在 master 上写监听服务，web 机器发送同步请求时，会将分组信息传到 master 上，然后重启 master，让其生效。

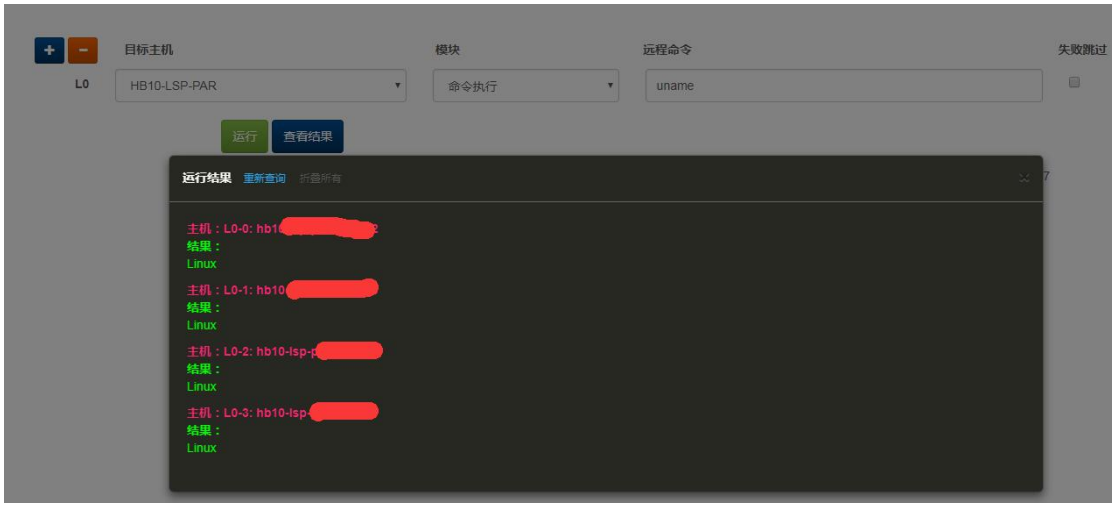
这是页面分组



节点和分组的关系是 manytomany 关系，一个节点可以存在于多个分组，一个分组可以拥有多个节点。分组完成后，要进入到执行阶段了，执行分为同步执行和异步执行、单机执行和组执行。这些 saltapi 里都有提供，自己封装一下即可。



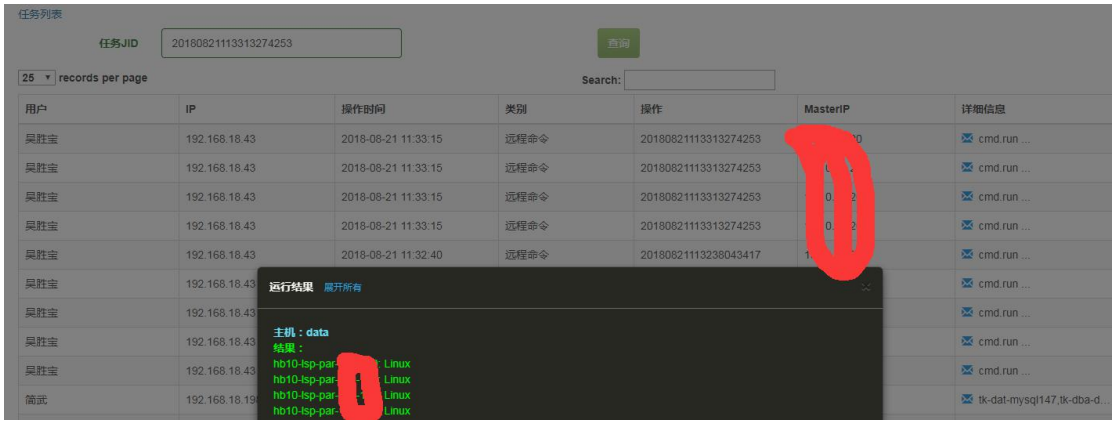
这是单机执行的页面



这是组执行的页面

模块部署

这是模块部署，部署原理就是上传相应的 sls 文件，然后调用 state.sls 执行即可，这点可以参照 saltapi 相关内容，比较简单。同时系统还会记录所有的操作记录，包括页面的操作记录和 saltstack 命令 的执行记录，方便以后回溯查询。

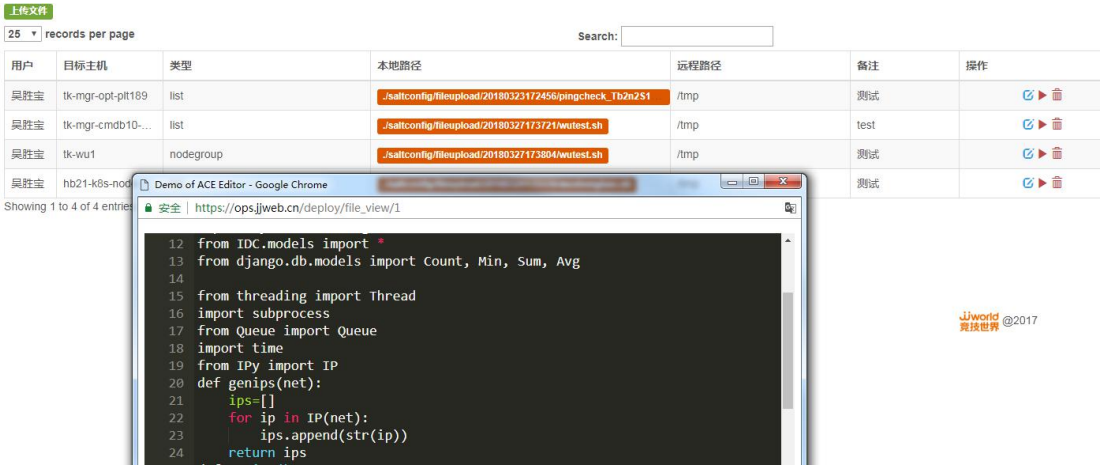


操作时间	2018-08-16 21:28:51
类别	更新Pillar
操作	更新Pillar
详细信息	hb30-dba-mysq BUSINESS: mysql DEPARTMENT: sscdba IDCNAME: HB30 IPADDR: 1 LOGTYPE: - lnxsys_cmdhis - mysql_error

记录操作时间，操作人，操作的节点以及操作的命令和结果。根据执行的 jid 可以回查结果，这个功能是 salt 自带的。

这是整个系统的一部分功能

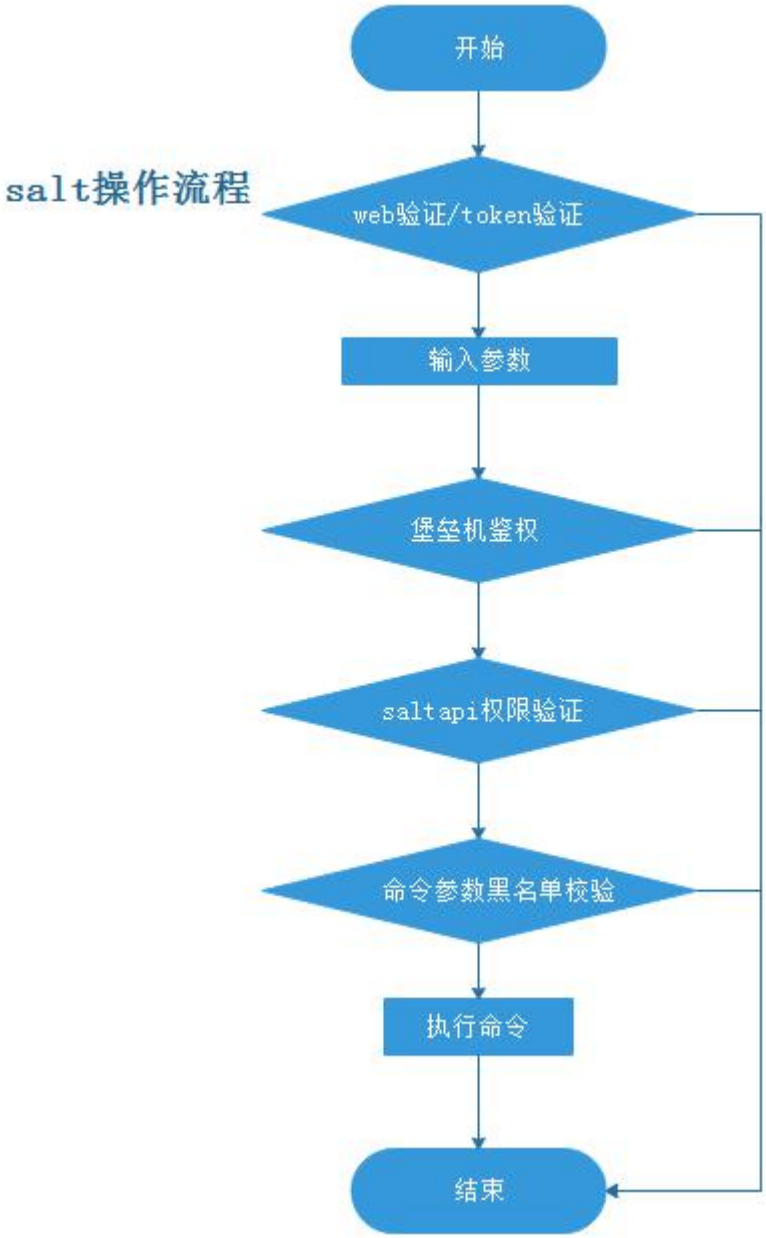
还有一个脚本上传预览和执行的功能



这是执行方面的一些功能，当然还有文件回滚等等功能目前还没有集成进来。

三、下面来说说安全方面做的一些工作

由于 salt 是批量操作，稍有不慎，会造成很大的影响，所以在执行之前，需要做一些限制。



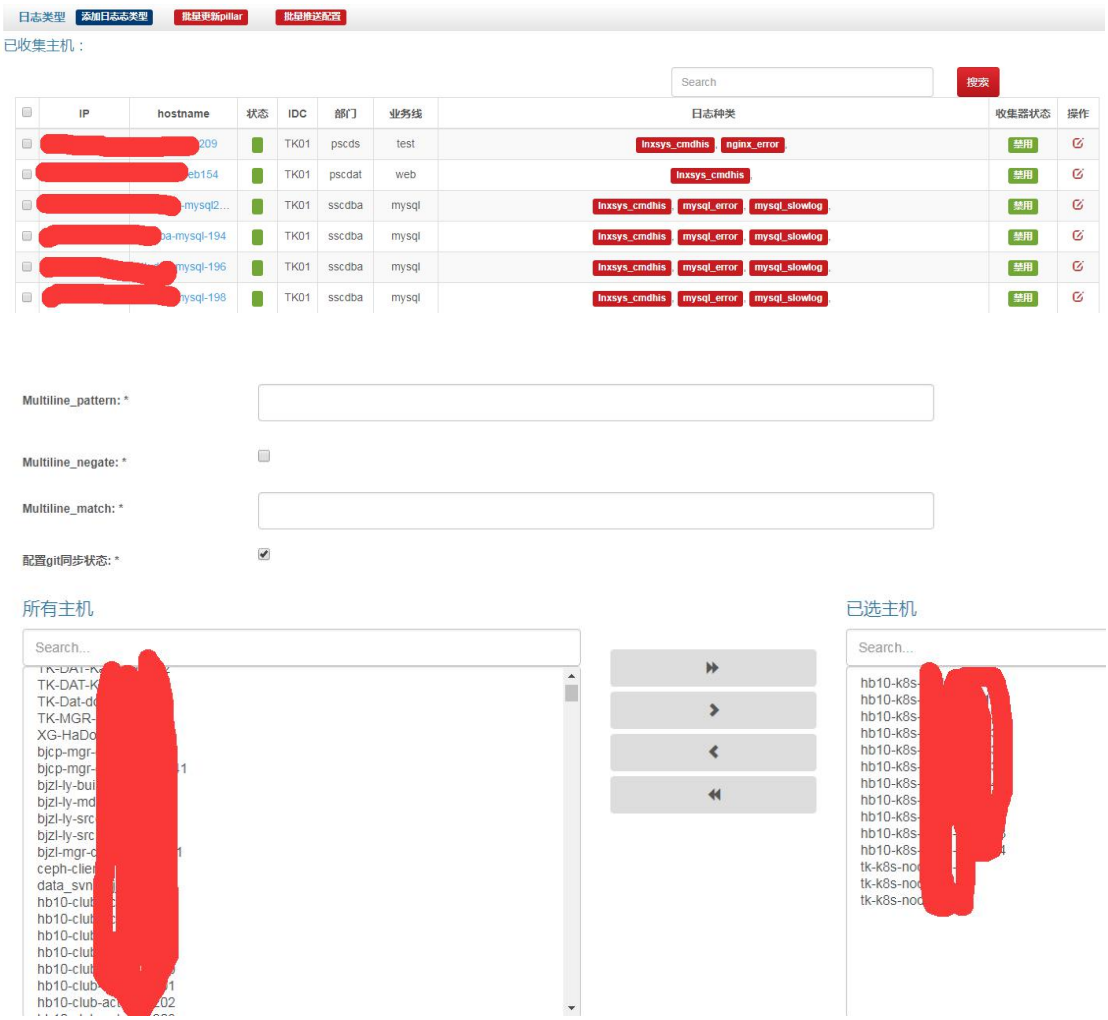
首先是 web 的页面验证，页面点击需要登录验证，api 调用需要 token 验证，保证随便就能操作的。这一步过去了，然后就验证操作人是否包含要操作的节点权限，具体做法是去堡垒机里面进行鉴权，每个用户在堡垒机里都会有权限列表，如果他要执行的节点没有权限就会直接打回请求。

```
47  print getresponse(url,token,{'username':username,'hostname':'zw-clu[redacted]','module':'cmd.run','cmd':'ifconfig','resul
48
49
50

{u'token': u'62e4b04d7c5858ddd80d4eeb9a34b6ec7dc3ce6e'}
wushengbao 您好:<br> 您没有zw-clu[redacted]机器的操作权限，如果需要权限请向系统部申请!
```

这样进一步限制了安全范围，操作人只能操作自己有权限的机器，其它机器无法执行任何命令。这两步过去了，还要对参数进行简单的检查，会设定一个命令黑名单，像 rm、shutdown、init、等敏感命令都无法执行，黑名单可以根据自己需要自行加减。

如图，我在黑名单里加了 uptime 命令，当我要执行的命令行里含有 uptime，就不会执行，会返回提示信息。以上是 salt 相关的一些操作的集成展示，下面说说其他的一些应用。



这两图是用 salt 来推送 elk 日志配置的功能页面。根据节点不同的日志类型，会生成不同的日志文件，并进行推送。日志类型和类型中的节点都可以自由分配。先是根据主要信息生成 pillar 配置，然后根据 pillar 配置往下推送，日志配置的 conf 信息放在 git 上。要推送的时候，slat 会从 git 上拉取。

最后就是 api 接口了，目前只提供了两种接口，一种 是执行接口，一种是根据 jid 查询结果的查询接口。执行接口根据参数设置分为同步执行和异步执行，参数为 sync 时会同步执行，直接返回结果，参数为 async 时会异步执行，只返回任务的 jid，然后用

户根据 jid 再调用查询接口去查询自己执行的任务结果。由于要经过层层校验，执行效率并不是太好，所以一般推荐用异步执行的方式，同时支持多个节点同时执行，api 不支持组执行，因为组权限不好控制。

以上就是我这次的主要分享内容了，欢迎探讨！谢谢！

ASK01:saltapi 走的 py 还是 http?

AN:应该是 https

ASK02:看图用的 vue 框架，又感觉页面 ui 有两个人系统?

AN:是两个.第一个是之前做的，没有用 vue.后来前后分离了，前端用 vue，后端还是 django.马上不搞 vue 了，准备转型 react 了.为了统一，可以寻求更多的技术支持.公司 web 都是用的 react，以后有复杂的功能页面可以使用他们的组件.或者直接寻求他们的支持.我们毕竟不是专业前端，复杂的的东西实现起来还是有点费力

ASK03:主机命名规范可以解读一下吗?

AN:我可以大概讲一下,因为这个都是部分很早就制定了.主机名规则是按照所在 idc、业务/部门、服务应用、ip 的后两位.目前只有物理服务器的主机名走 dns，映射的也是管理 ip，而不是业务 ip.命名规范因公司而异.

ASK:Django Token 走的 JWT 吗?

AN:像分组配置的同步和 master 的重启都是走的 rpc.我用的是 rypc 模块.没有，目前就是用的最基本的 token 验证，只是限制了一下 ip 和访问频次.