

# 设备初始化后端 API 文档说明

Ansible 是作为自动化运维的底层实现，功能很强大，但需要通过命令或 playbook 的 yaml 文件来实现，相对对运维人员而言，学习成本过大。所以这里要考虑到通过 Flask Web 框架来实现其二次封装，提供 HTTP 接口来实现远程调用。但我们在请求 Ansible API 的时候，ansible 默认本身是阻塞的，用户那边会一直处于等待状态，这样大家的用户体验也不好，所以这里会用 rq 来实现其非阻塞功能，即实现任务的异步化。

目前在 BOSS 界面前端可以实现初始化操作，如下图所示：

初始化待选设备													
主机名	设备型号	合同编号	SN号	机房	机柜	IP	平台	设备状态	上架时间	最后更新时间	Fwsw	操作	
	曙光天韵620-G10	MA-34197-4	9800069400506664	浙江金华电信--三路口	ZJ-JH(CTL)-10-09#	111.1.9.114	DCBG-Test	未初始化	2018-06-04	2018-06-04	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506664	浙江金华电信--三路口	ZJ-JH(CTL)-10-09#	111.1.9.115	DCBG-Test	未初始化	2018-06-04	2018-06-04	未安装	<a href="#">🔗</a>	
	PowerEdge R730xd		exehtp57IE9	湖南广电机房	机柜1	119.44.17.244		未初始化	2018-05-04	2018-05-04	未安装	<a href="#">🔗</a>	
	PowerEdge R730xd		sM245lindz	湖南广电机房	机柜1	119.44.17.245		未初始化	2018-05-04	2018-05-04	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506480	江苏镇江电信	JS-ZJ-G-11# (电信)	222.186.20.38	vps.realhost	未初始化	2018-01-12	2018-01-12	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506480	江苏镇江电信	JS-ZJ-G-11# (电信)	222.186.20.39	vps.realhost	未初始化	2018-01-12	2018-01-12	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506480	江苏镇江电信	JS-ZJ-G-11# (电信)	222.186.20.37	vps.realhost	未初始化	2018-01-12	2018-01-12	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506480	江苏镇江电信	JS-ZJ-G-11# (电信)	222.186.20.36	vps.realhost	未初始化	2018-01-12	2018-01-12	未安装	<a href="#">🔗</a>	
ctl-zj-183-131-159-008	曙光天韵620-G10	MA-34197-4	9800069400506664	浙江金华电信--三路口	ZJ-JH(CTL)-10-09#	183.131.159.8	akamai-license	未初始化	2018-01-09	2018-01-09	未安装	<a href="#">🔗</a>	
oth-bj-119-090-062-185	曙光天韵620-G10	MA-34197-3	9800069400506688	北京数北机房	BJ-SB-Y-3	119.90.62.185	stepping.server	未初始化	2017-12-13	2017-12-13	安装	<a href="#">🔗</a>	
cnc-hn-210-022-063-204	曙光天韵620-G10	MA-34197-4	9800069400506477	湖南长沙联通	HN-CS-C20#	210.22.63.204	stepping.server	未初始化	2017-12-13	2018-04-17	安装	<a href="#">🔗</a>	
ctl-fj-218-006-023-028	曙光天韵620-G10	MA-34197-4	9800069400506485	福建莆田电信--东刺路	FJ-PT-B3-4#	218.6.23.28	c01J05	未初始化	2017-12-05	2018-04-16	未安装	<a href="#">🔗</a>	
ctl-fj-218-006-023-029	曙光天韵620-G10	MA-34197-4	9800069400506485	福建莆田电信--东刺路	FJ-PT-B3-4#	218.6.23.29	c01J05	未初始化	2017-12-05	2018-04-16	未安装	<a href="#">🔗</a>	
ctl-fj-218-006-023-030	曙光天韵620-G10	MA-34197-4	9800069400506485	福建莆田电信--东刺路	FJ-PT-B3-4#	218.6.23.30		未初始化	2017-12-05	2018-04-16	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506485	福建莆田电信--东刺路	FJ-PT-B3-4#	218.6.23.31	vps.pk	未初始化	2017-12-05	2018-04-16	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506485	福建莆田电信--东刺路	FJ-PT-B3-4#	218.6.23.32	vps.idc-test	未初始化	2017-12-05	2018-04-16	未安装	<a href="#">🔗</a>	
oth-a_na-172-31-23-90	SuperServer SU128			海外	AWS-亚太区域(新加坡)	52.221.243.226	overseas	未初始化	2017-11-30	2017-11-30	安装	<a href="#">🔗</a>	
ctl-fj-110-080-134-149	曙光天韵620-G10	MA-34197-4	9800069400506692	福建厦门电信--软件园2期	FJ-XM-3F-IDF2-14#	110.80.134.149	vps.idc-test	未初始化	2017-11-02	2018-01-03	未安装	<a href="#">🔗</a>	
	曙光天韵620-G10	MA-34197-4	9800069400506676	浙江金华电信--三路口	ZJ-JH-09-08# (米粒)	61.130.28.223	vps.idc-test	未初始化	2017-11-02	2018-04-19	未安装	<a href="#">🔗</a>	
ctl-zj-115-231-047-102	曙光天韵620-G10	MA-31364	9800069400506675	浙江宁波移动鄞州机房 (新)	ZJ-NB-F-4 (移动)	115.231.47.102	vps.idc-test	未初始化	2017-11-02	2017-11-02	未安装	<a href="#">🔗</a>	
	SuperServer SU128			青海移动机房	1#	111.44.243.102		未初始化	2017-10-31	2017-10-31	未安装	<a href="#">🔗</a>	

## 步骤一 勾选要进行设备初始化的机器



## 步骤二 选中执行“设备初始化”菜单，点确认

## 设计思路

前端的工作：

1、前端输入命令；

- 
- 2、JS 将前端命令输入发送到后端；
  - 3、后接收前端的输入，执行并返回任务 ID；
  - 4、前端 JS 通过任务 ID 去后端获取任务结果。

后端的工作：

- 1、通过路由将制定 URL 的请求转发到相应的函数上；
- 2、相应的函数将前端请求发送到 Ansible，Ansible 执行 playbook task 任务，并获得返回值；
- 3、对返回值进行相应的处理并返回到前端。

## 项目需求背景

一、全部运维操作尽量 Web 化实现，此项操作要求没有脚本能力的产品部门、资产部门及运维监控同事能执行“设备初始化”功能。

二、设备初始化机器目前在预装系统之后（用公司封装好的镜像，里面封装好了跳板机的公钥及基础的安全运维方面），SSH 协议 是开放的；在几款自动化配置管理工具，如：Puppet、FabricAnsible 及 Saltstack 中选型，经过综合评估，最终考虑了 Ansible（不需要配置 Client 客户端）。

## 使用开源软件说明

## 版本

Python 版本为 2.7.9

Ansible 版本为 1.9.6

## 使用的开源软件

Ansible

Flask

redis

redis-queue

## 正式版本 v0.1

## 项目文件明细

inventoryv0.1.py .... 正式的功能文件，版本为 v0.1

mytest.yml..... 用来实现初始化功能的 ansible playbook 的 YAML 文件，主要用于初始化机器

---

的业务初始化等。

somejob.py..... 用 Flask 来封装 ansible API，使用方法为 get（测试用）

somejobv2.py .... 实现了动态 inventofy 功能，使用的方法为 POST（测试用）

somejobv2.1.py ... 将前端提交的数据格式由 Form 改为 json 格式（测试用）

work.py..... 用 rq 来实现任务异步化（非阻塞）。

client.py... 客户端测试脚本，不过感觉 postman 测试起来更加方便。

前期考虑用 Celery 框架来实现异步非阻塞的功能，但在实际使用及学习过程中发现使用较复杂，学习成本较大，改用更轻量级的 rq 来实现需求。

首先基于安全的考虑，暂时只允许武汉办公区域的 IP 地址及 Boss 前端访问此 API（其它访问 5000 端口的请求全部拒绝），如下所示：

```
iptables -I INPUT -p TCP --dport 5000 -j DROP
iptables -I INPUT -s 113.57.171.138 -p TCP --dport 5000 -j ACCEPT
iptables -I INPUT -s 59.173.242.206 -p TCP --dport 5000 -j ACCEPT
iptables -I INPUT -s 124.251.24.203 -p TCP --dport 5000 -j ACCEPT
iptables -I INPUT -s 124.243.198.170 -p TCP --dport 5000 -j ACCEPT
```

## 启动步骤

1.先启动 redis-server，为了安全起见，只对 127.0.0.1 开放。

```
/usr/local/redis/bin/redis-server /usr/local/redis/etc/redis.conf
```

2.启动 initial.py 程序，开启 Flask 应用封装 Ansible API。

```
nohup /usr/local/bin/python initialv0.1.py &
```

3.启动 work 程序，开启 rq 队列任务。

---

```
nohup /usr/local/bin/python work.py &
```

4.我们可以在别的机器上执行 POST 请求，以 http 方式来执行 Ansible playbook 任务了，API 接口为：

```
http://202.85.220.118:5000/ansible/playbook/
```

此处已跟前端团队沟通，需要初始化的设备 IP 列表以 Form 或 Json 的格式（推荐 Json）的格式传递，我在 Flask Web 里面用了 **jinja2 渲染成 Ansible 能识别的格式**，下面是前端传递的例子：

```
{
    "ips":
    [
        "183.131.161.37",
        "183.213.22.21",
        "139.214.115.186"
    ]
}
```

此段 http 就可以执行 hosts 为 initial[前端提供的需要初始化的 IP 机器列表], 另外名字为 mytest 的 Ansible Playbook 了，里面可以执行我们的初始化脚本。

后端接口说明（此部分提供给前端同事，方便工作对接）

## 1、Request 要求

URI: http://202.85.220.118:5000/ansible/playbook/

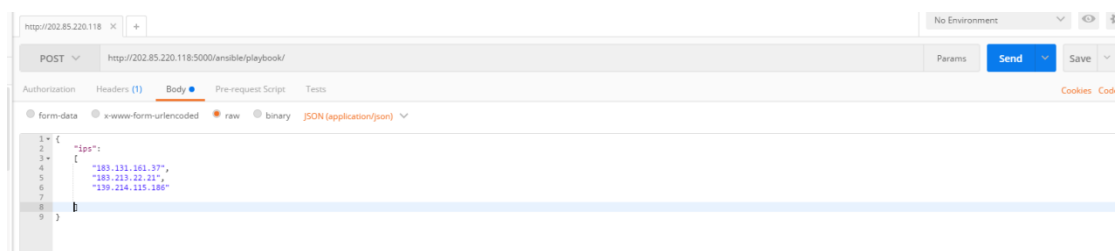
Method: POST

格式为: json

其中 `ips` 为必须带的值, []里面输入单个 IP 或多个 IP 的集合, 格式如下:

```
{
  "ips":
  [
    "183.131.161.37",
    "183.213.22.21",
    "139.214.115.186"
  ]
}
```

截图如下所示:



基本安全的原因考虑, 目前只允许武汉办公区域的公网 IP 及前端 IP 进行访问此 API, 注意安全!

执行完成以后返回异步的 `task_id` 值, 如下所示:

```
5c5b6036-3ca9-4992-80c8-ab54564b9536
```

## 2.Response 说明

URI: <http://202.85.220.118:5000/ansible/results/>

---

Method: GET

接口后面跟上上面异步返回的 `task_id` 值，例如：

```
http://202.85.220.118:5000/ansible/results/4366c076-dfe0-402e-9cd5-a2130906ea69
```

这样就可以取得 Ansible 执行任务后的完整的返回结果了，然后根据其结果来判断任务是否成功或失败。

项目地址放于自己的 GitLab 里了，大家有兴趣可以下载研究，如下所示：

```
https://github.com/yuhongchun/devops/tree/master/ansible
```