

# grains

2016年1月28日 17:14

整理：

cpu	cpu_model	获取型号，主频，制造商	
	num_cpus	获取逻辑核心数	
内存	mem_total	总内存大小	需要补充内存插槽数量/内存单条容量
网卡	hwaddr_interfaces	获取网卡名称和对应的mac	
	ip_interfaces	获取网卡名称和对应的ip	
虚拟化信息	virtual	获取是否是物理机器physical/xen	
整机信息	machine_id	唯一ID	
	serialnumber	服务编号	虚拟机不存在该值，另外需要补充快速服务代码
	server_id	id，不建议采用该值	
	manufacturer	制造商	虚拟机不存在该值
	productname	服务器型号	虚拟机不存在该值
bios	biosversion	bios的版本	虚拟机不存在该值
	biosreleasedate	bios的日期	虚拟机不存在该值
			缺少硬盘，raid信息，插槽数
os	os	分支名，比如centos	
	osarch	显示系统位数，比如x86_64	
	osrelease	os版本号，比如6.6	
saltminion	saltversion	显示salt-minion的版本号	
python	pythonexecutable	显示python的可执行路径	
	pythonversion	显示版本信息	
内核	kernelrelease	内核版本	
主机名	fqdn	主机名	
环境变量	locale_info	语言，编码信息	
selinux	selinux	selinux开关状态	

需要编写插件补充以下标粗信息

非标粗信息为动态信息，需要使用其他方式采集

1，服务器作用，位置信息，负责人，联系方式，保修信息，供货商信息（这个手工录入就行了）

2，安全信息：

**系统启动时间/已运行时间**

**用户数/当前登录用户**

**防火墙是否开启**

3, 状态信息 ( 以下动态信息建议其他工具监控 ) :

内存使用量/swap使用量

cpu负载

磁盘io

**分区信息 ( 分区名称 , 总容量 , 分区格式 )**

网卡流量信息

**其他软件信息 ( 比如是否运行了redis , redis端口是什么 , 比如是否运行了mysql , 是否运行了nginx等 )**

grains编写示例如下 :

创建文件/srv/salt/\_grains/ops\_mem.py

---

```
# -*- coding: utf-8 -*-

'''
Module for squid disk information by python
'''

import commands
import os

def cache():
    '''
    Return the memory usage information for volumes mounted on this minion
    '''
    grains={}
    m = commands.getoutput("free -m|awk '$0~/Mem/ {print$2}'")
    grains['all_mem']=int(m)
    n = commands.getoutput("free -m|awk '$2~/buffers/ {print$3}'")
    grains['used_mem']=int(n)
    return grains
```

---

以上脚本定义了all\_mem和used\_mem两个字段

然后执行salt test75 saltutil.sync\_all将模块同步到指定的minion节点

执行salt test75 sys.reload\_modules让节点重新加载模块

使用以下命令即可获取给定的值

salt test75 grains.item all\_mem