**1、 文件组装模块-assemble**

assemble主要是将多份配置文件组装为一份配置文件。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 必填 | 默认 | 选项 | 说明 |
| Backup | 否 | No | Yes/no | 是否创建备份文件，使用时间戳 |
| Delimiter | 否 |  |  | 配置文件内容之间的分隔符 |
| Dest | 是 |  |  | 目的文件用来连接所有的源文件 |
| Follow | 否 | No | Yes/no | 表示文件系统的链接，如果存在，应该follow |
| Group | 否 |  |  | 文件所属组，相当于chown |
| Mode | 否 |  |  | 文件属性，0644，记住保留前面的0 |
| Owner | 否 |  |  | 文件所属用户 |
| Regexp | 否 |  |  | 组装的时候是否匹配相关的文件进行组合 |
| Remote\_src | 否 | True | True/False | False表示src文件在本机上，True表示src在远程主机上 |
| Src | 是 |  |  | 已经存在的目录文件 |

**1.1 将本机上的配置文件组装发送到远程主机**

|  |
| --- |
| [root@ansibleserver kel]# ansible pythonserver -m assemble -a "src=/root/kel/ dest=/tmp/kel.conf remote\_src=False"  SSH password:  192.168.1.60 | success >> {  "changed": true,  "dest": "/tmp/kel.conf",  "gid": 0,  "group": "root",  "md5sum": "23245453347840c2aa288bcc274bc298",  "mode": "0600",  "owner": "root",  "size": 32,  "src": "/root/.ansible/tmp/ansible-tmp-1453144282.9-4482221307096/src",  "state": "file",  "uid": 0  } |

在本机目录/root/kel的文件进行组装，发送到远程配置文件/tmp/kel.conf中，在此注意参数remote\_src为False表示组装本机的配置文件

如果在本机目录中不存在文件，那么将会显示[Python](http://lib.csdn.net/base/11)代码和乱码

**1.2 使用分隔符**

|  |
| --- |
| [root@ansibleserver kel]# ansible pythonserver -m assemble -a "src=/root/kel/ dest=/tmp/kel.conf remote\_src=False delimiter='####'"  SSH password:  192.168.1.60 | success >> {  "changed": true,  "dest": "/tmp/kel.conf",  "gid": 0,  "group": "root",  "md5sum": "b46dcc49d69688a6be5bfd5b82737c0c",  "mode": "0600",  "owner": "root",  "size": 42,  "src": "/root/.ansible/tmp/ansible-tmp-1453144492.01-279709420329993/src",  "state": "file",  "uid": 0  } |

使用分隔符进行分割每个配置文件的内容，分隔符参数为delimiter

**2、 文件复制模块-copy**

copy模块是将本机中的文件复制到远程主机当中，fetch模块是将远程主机中的内容复制到本机中，如果在复制文件的过程当中需要使用变量，那么就使用template模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 必填 | 默认 | 选项 | 说明 |
| Backup | 否 | No | Yes/no | 是否创建备份文件，使用时间戳 |
| Content | 否 |  |  | 当使用content替代src的时候，可以把文档中内容设定为特定值 |
| Dest | 是 |  |  | 文件复制的目的地 |
| Follow | 否 | No | Yes/no | 表示文件系统的链接，如果存在，应该follow |
| Group | 否 |  |  | 文件所属组，相当于chown |
| Mode | 否 |  |  | 文件属性，0644，记住保留前面的0 |
| Owner | 否 |  |  | 文件所属用户 |
| Directory\_mode | 否 |  |  | 设定目录的权限，只有在新建的时候才会使用，不会影响已经存在的目录 |
| Remote\_src | 否 | True | True/False | False表示src文件在本机上，True表示src在远程主机上 |
| Src | 否 |  |  | 已经存在的目录文件 |
| Force | 否 | No | Yes/no | 是否覆盖 |
| Validate | 否 |  |  | 复制前是否需要检验复制的目的地 |

**2.1 设定权限进行拷贝**

|  |
| --- |
| [root@ansibleserver kel]# ansible pythonserver -m copy -a "src=/root/kel/1 dest=/tmp/kel owner=root group=root mode=0777"  SSH password:  192.168.1.60 | success >> {  "changed": true,  "dest": "/tmp/kel",  "gid": 0,  "group": "root",  "md5sum": "799e373c68e4b72dd2eb7ed715cfced9",  "mode": "0777",  "owner": "root",  "size": 4,  "src": "/root/.ansible/tmp/ansible-tmp-1453145233.73-276909183600356/source",  "state": "file",  "uid": 0  } |

设定了文件的权限，设定mode为0777，用户和组均为root

**2.2 拷贝的时候备份**

|  |
| --- |
| [root@ansibleserver kel]# ansible pythonserver -m copy -a "src=/root/kel/1 dest=/tmp/kel owner=root group=root backup=yes"  SSH password:  192.168.1.60 | success >> {  "backup\_file": "/tmp/kel.2016-01-18@14:34~",  "changed": true,  "dest": "/tmp/kel",  "gid": 0,  "group": "root",  "md5sum": "a8cb3ba3ab8a8840e019c6b21521c483",  "mode": "0777",  "owner": "root",  "size": 8,  "src": "/root/.ansible/tmp/ansible-tmp-1453145701.59-117060156554446/source",  "state": "file",  "uid": 0  } |

备份的时候，只有在文件发生了变化的时候，那么会在远程主机上进行备份，而不是在本机上进行备份源文件，备份的是远程主机上被修改的文件

**2.3 拷贝文件之后进行验证**

|  |
| --- |
| [root@ansibleserver kel]# ansible pythonserver -m copy -a "src=/etc/sudoers dest=/tmp/2 validate='visudo -cf %s'"  SSH password:  192.168.1.60 | success >> {  "changed": true,  "dest": "/tmp/2",  "gid": 0,  "group": "root",  "md5sum": "4093e52552d97099d003c645f15f9372",  "mode": "0644",  "owner": "root",  "size": 3797,  "src": "/root/.ansible/tmp/ansible-tmp-1453146150.33-246902811238377/source",  "state": "file",  "uid": 0  } |

拷贝之后进行验证

**3、 文件拉取模块-fetch**

文件拉取模块主要是将远程主机中的文件拷贝到本机中，和copy模块的作用刚刚相反，并且在保存的时候使用hostname来进行保存，当文件不存在的时候，会出现错误，除非设置了选项fail\_on\_missing为yes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 必填 | 默认值 | 选项 | 说明 |
| Dest | Yes |  |  | 用来存放文件的目录，例如存放目录为backup，源文件名称为/etc/profile在主机pythonserver中，那么保存为/backup/pythonserver/etc/profile |
| Fail\_on\_missing | No | No | Yes/no | 当源文件不存在的时候，标识为失败 |
| Flat | No |  |  | 允许覆盖默认行为从hostname/path到/file的，如果dest以/结尾，它将使用源文件的基础名称 |
| Src | Yes |  |  | 在远程拉取的文件，并且必须是一个file，不能是目录 |
| Validate\_checksum | No | Yes | Yes/no | 当文件fetch之后进行md5检查 |

**3.1 fetch一个文件进行保存**

|  |
| --- |
| [root@ansibleserver ~]# ansible pythonserver -m fetch -a "src=/root/123 dest=/root"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "dest": "/root/192.168.1.60/root/123",      "md5sum": "31be5a34915d52fe0a433d9278e99cac",      "remote\_md5sum": "31be5a34915d52fe0a433d9278e99cac"  } |

src表示为远程主机上需要传送的文件路径，dest表示为本机上的路径，在传送过来的文件，是按照IP地址进行分类，然后路径是源文件的路径

在拉取文件的时候，必须拉取的是文件，不能拉取文件夹

**3.2 指定路径目录进行保存**

|  |
| --- |
| [root@ansibleserver ~]# ansible pythonserver -m fetch -a "src=/root/Ssh.py dest=/root/kel/ flat=yes"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "dest": "/root/kel/Ssh.py",      "md5sum": "63f8a200d1d52d41f6258b41d7f8432c",      "remote\_md5sum": "63f8a200d1d52d41f6258b41d7f8432c"  } |

在使用参数为flat的时候，如果dest的后缀名为/，那么就会保存在目录中，然后直接保存为文件名；当dest后缀不为/的时候，那么就会直接保存为kel的文件，如上例所示。

主要是在于dest是否已/结尾，从而来区分这是个目录还是路径。

**4、 文件管理模块-file**

主要用来设置文件、链接、目录的属性，或者移除文件、链接、目录，很多其他的模块也会包含这种作用，例如copy，assemble和template

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 必填 | 默认 | 选项 | 说明 |
| Follow | No | No | Yes/no | 这个标识说明这是系统链接文件，如果存在，应该遵循 |
| Force | No | No | Yes/no | 强制创建链接在两种情况下：源文件不存在（过会会存在）；目标存在但是是文件（创建链接文件替代） |
| Group | No |  |  | 文件所属用户组 |
| Mode | No |  |  | 文件所属权限 |
| Owner | No |  |  | 文件所属用户 |
| Path | Yes |  |  | 要控制文件的路径 |
| Recurse | No | No | Yes/no | 当文件为目录时，是否进行递归设置权限 |
| Src | No |  |  | 文件链接路径，只有状态为link的时候，才会设置，可以是绝对相对不存在的路径 |
| State | No | File | File/link  Directory  Hard/touch  Absent | 如果是目录不存在，那么会创建目录；如果是文件不存在，那么不会创建文件；如果是link，那么软链接会被创建或者修改；如果是absent，那么目录下的所有文件都会被删除，如果是touch，会创建不存在的目录和文件 |

**4.1 设置文件属性**

|  |
| --- |
| [root@ansibleserver ~]# ansible pythonserver -m file -a "path=/root/123 owner=kel group=kel mode=0644"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "gid": 500,      "group": "kel",      "mode": "0644",      "owner": "kel",      "path": "/root/123",      "size": 294,      "state": "file",      "uid": 500  } |

文件路径为path，表示文件路径，设定所属用户和所属用户组，权限为0644

|  |
| --- |
| [root@ansibleserver ~]# ansible pythonserver -m file -a "path=/tmp/kel/ owner=kel group=kel mode=0644 recurse=yes"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "gid": 500,      "group": "kel",      "mode": "0644",      "owner": "kel",      "path": "/tmp/kel/",      "size": 4096,      "state": "directory",      "uid": 500  } |

文件路径为path，使用文件夹进行递归修改权限，使用的参数为recurse表示为递归

**4.2 创建目录**

|  |
| --- |
| [root@ansibleserver ~]# ansible pythonserver -m file -a "path=/tmp/kel state=directory mode=0755"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "gid": 0,      "group": "root",      "mode": "0755",      "owner": "root",      "path": "/tmp/kel",      "size": 4096,      "state": "directory",      "uid": 0  } |

创建目录，使用的参数主要是state为directory

4.3 修改权限

|  |
| --- |
| [root@ansibleserver ~]# ansible pythonserver -m file -a "path=/tmp/kel mode=0444"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "gid": 0,      "group": "root",      "mode": "0444",      "owner": "root",      "path": "/tmp/kel",      "size": 4096,      "state": "directory",      "uid": 0  } |

直接使用mode来进行修改权限

4.4 创建软连接

|  |
| --- |
| [root@ansibleserver tmp]# ansible pythonserver -m file -a "src=/tmp/1 dest=/tmp/2 owner=kel state=link"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "dest": "/tmp/2",      "gid": 0,      "group": "root",      "mode": "0777",      "owner": "kel",      "size": 6,      "src": "/tmp/1",      "state": "link",      "uid": 500  } |

 src表示已经存在的文件，dest表示创建的软连接的文件名，最后的state状态为link

**5、 ini文件管理模块-ini\_file**

Ini\_file模块主要是用来设置ini文件的格式的文件。

添加、删除、修改单独的ini格式的文件，其中主要是用来修改部分的配置信息，而不是template或者是assemble，添加失去的section。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 必填 | 默认 | 选择 | 说明 |
| Backup | No | No | Yes/no | 在修改之前进行备份 |
| Dest | Yes |  |  | Ini类型文件的路径，如果不存在会创建 |
| Follow | No | No | Yes/no | 标识是否文件系统链接，如果存在，那么将遵循 |
| Group | No |  |  | 组名 |
| Mode | No |  |  | 权限 |
| Option | No |  |  | 如果需要修改一个值，那么可以设置，如果添加删除一个整段，那么可以忽略 |
| Others | No |  |  | 文件模块所使用的参数可以在这里使用 |
| Owner | No |  |  | 用户名 |
| Section | Yes |  |  | 在ini文件中的section名称，如果状态为present并且设置了一个值，那么会添加 |
| State | No | Present | Absent  Present | 新增/删除 |
| Value | No |  |  | 这个值和option相关联，如果删除那么会忽略 |

**5.1 添加其中的节的值**

|  |
| --- |
| [root@ansibleserver tmp]# ansible pythonserver -m ini\_file -a "dest=/tmp/kel section=kel option=kel value=kel mode=0600 backup=yes"  SSH password:  192.168.1.60 | success >> {      "changed": true,      "dest": "/tmp/kel",      "gid": 0,      "group": "root",      "mode": "0600",      "msg": "OK",      "owner": "root",      "size": 29,      "state": "file",      "uid": 0  } |

表示在ini文件格式中添加一个值，路径为/tmp/kel，节名称为kel，字符为kel值为kel，在修改之前进行备份，ini文件格式如下：

[section]

key=value

标签: [ansible](http://www.cnblogs.com/zenghui940/tag/ansible/)