**1 案例1：配置sudo权限**

**1.1 问题**

本案例要求使用sudo提升普通用户的权限，要求如下：

* 给所有被管理主机创建系统账户
* 账户名称为alice，密码为123456
* 修改sudo配置，让alice可以执行任何管理命令

**1.2 方案**

sudo（superuser or another do）让普通用户可以以超级管理员或其他人的身份执行命令。

sudo基本流程如下：

1. 管理员需要先授权（修改/etc/sudoers文件）
2. 普通用户以sudo的形式执行命令

修改/etc/sudoers的方法如下：

1. visudo（带语法检查，默认没有颜色提示）
2. vim /etc/sudoers（不带语法检查，默认有颜色提示）

授权格式如下：

用户或组 主机列表=(提权身份) [NOPASSWD]:命令列表

注意事项：命令需要写绝对路径，对组授权需要在组名称前面加%。

1. **[**root@control **~]**# cat **/**etc**/**sudoers #不要改，下面仅仅是语法格式的示例（例子）
2. … …
3. root ALL**=(**ALL**)** ALL
4. tom ALL**=(**root**)** **/**usr**/**bin**/**systemctl
5. **%**wheel      ALL**=(**ALL**)** ALL

**1.3 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：配置sudo提权**

1）远程所有被管理主机批量创建系统账户，账户名称为alice，密码为123456。

1. **[**root@control ansible**]**# ansible all **-**m user **-**a "name=alice \
2. password={{'123456' | password\_hash('sha512')}}"

2）配置alice账户可以提权执行所有命令（control批量授权，node1主机验证）。

使用lineinfile模块修改远程被管理端主机的/etc/sudoers文件，line=后面的内容是需要添加到文件最后的具体内容。

等于是在/etc/sudoers文件末尾添加一行:alice ALL=(ALL) NOPASSWD:ALL

1. **[**root@control ansible**]**# ansible all **-**m lineinfile **\**
2. **-**a "path=/etc/sudoers line='alice ALL=(ALL) NOPASSWD:ALL'"
3. **[**root@control **~]**# ssh alice@node1

如何验证？可以在node1电脑上面使用alice用户执行sudo重启服务的命令看看是否成功。

1. **[**alice@node1 **~]**$ sudo systemctl restart sshd #不需要输入密码

**2 案例2：修改Ansible配置**

**2.1 问题**

沿用练习一，修改ansible配置实现使用普通用户远程被控制端主机，具体要求如下：

* 修改主配置文件
* 设置ansible远程被管理端主机账户为alice
* 设置ansible远程管理提权的方式为sudo
* 修改主机清单文件
* 修改主机清单配置文件，添加SSH参数

**2.2 步骤**

实现此案例需要按照如下步骤进行。

**步骤一：配置普通用户远程管理其他主机**

1）修改主配置文件，配置文件文件的内容可以参考/etc/ansible/ansible.cfg。

1. **[**root@control **~]**# vim **~**/ansible/ansible**.**cfg
2. **[**defaults**]**
3. inventory **=** **~**/ansible/inventory
4. remote\_user **=** alice                #以什么用户远程被管理主机（被管理端主机的用户名）
5. **[**privilege\_escalation**]**
6. become **=** **true**                    #alice没有特权，是否需要切换用户提升权限
7. become\_method **=** sudo                #如何切换用户（比如用su就可以切换用户，这里是sudo）
8. become\_user **=** root                #切换成什么用户（把alice提权为root账户）
9. become\_ask\_pass **=** no                #执行sudo命令提权时是否需要输入密码

思考：

如果A主机ssh远程访问B主机，应该输入哪个主机的用户名和对应的密码？

如果张三要去李四家，应该使用谁家的钥匙，打开谁家的门？

2)远程被管理端主机的alice用户，需要提前配置SSH密钥。

1. **[**root@control **~]**# **for** i **in** node1 node2 node3 node4 node5
2. **do**
3. ssh**-**copy**-**id alice@$i
4. done

验证效果：

1. **[**root@control **~]**# ssh alice@node1 #依次远程所有主机看看是否需要密码
2. #注意：是远程登录node1，应该输入的是node1电脑上面alice账户的密码，control没有alice用户
3. **[**root@control **~]**# ansible all **-**m command **-**a "who" #测试效果
4. **[**root@control **~]**# ansible all **-**m command **-**a "touch /test" #测试效果

常见报错（有问题可以参考，没问题可以忽略）：

1. node1 **|** UNREACHABLE**!** **=>** **{**
2. "changed"**:** **false,**
3. "msg"**:** "Failed to connect to the host via ssh: alice@node1: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password)."**,**
4. "unreachable"**:** **true**
5. **}**
6. 问题分析：
7. 英语词汇：Failed（失败），connect（连接），to（到），host（主机），via（通过）
8. permission（权限），denied（被拒绝）
9. Failed to connect to host via ssh alice@node1（通过ssh使用alice远程连接到主机失败）
10. Permission denied（因为无法连接，所以报错说权限被拒绝）
11. 解决办法：手动ssh alice@主机名（如node1），看看是否可以实现免密码登录。
12. Ansible的原理是基于ssh远程管理，如果无法实现alice免密码登录，则实验会失败！
13. 如何实现免密码登录，可以参考案例上面的命令，或者第一阶段相关知识。

3）修改inventory主机清单配置文件（参考即可，不需要操作）。

如果个别主机的账户不同，该如何处理呢？

如果有些主机需要使用密码远程呢？如果有些主机的SSH端口不是22呢？

1. **[**root@control **~]**# cat **~**/ansible/inventory
2. **[**test**]**
3. node1     ansible\_ssh\_port**=220**             #自定义远程SSH端口
4. **[**proxy**]**
5. node2 ansible\_ssh\_user**=**alice                    #自定义远程连接的账户名
6. **[**webserver**]**
7. node**[3:4]** ansible\_ssh\_pass**=**密码                 #自定义远程连接的密码
8. **[**database**]**
9. node5
10. **[**cluster**:**children**]**
11. webserver
12. database

**3 案例3：Playbook应用案例**

**3.1 问题**

沿用练习二，编写Ansible Playbook剧本，使用Playbook完成自动化操作，具体要求如下：

* 熟悉Playbook语法格式
* 编写Playbook管理系统账户
* 编写Playbook管理逻辑卷
* 编写Playbook管理软件包

**3.2 方案**

Ansible ad-hoc可以通过命令行形式远程管理其他主机，适合执行一些临时性简单任务。另外还有一种远程管理的方式叫Playbook，Ansible Playbook中文名称叫剧本,它将经常需要执行的任务写入一个文件，这个文件就叫剧本。

* 剧本中可以包含多个任务
* 剧本写后，我们随时根据剧本，执行相关的任务命令
* Playbook剧本要求按照YAML格式编写
* 适合执行周期性经常执行的复杂任务

YAML是什么？

* YAML是一个可读性高、用来表达数据序列的格式语言
* YAML：YAML Ain't a Markup Language
* YAML以数据为中心，重点描述数据的关系和结构

YAML的格式要求如下：

* "#"代表注释，一般第一行为三个横杠（---）
* 键值（key/value）对使用":"表示，数组使用"-"表示
* key和value之间使用":"分隔
* ":"后面必须有空格 “-”后面也必须有空格
* 一般缩进由两个或以上空格组成
* 相同层级的缩进必须对齐，缩进代表层级关系
* 全文不可以使用tab键
* 区分大小写
* 扩展名为yml或者yaml
* 跨行数据需要使用>或者|，其中|会保留换行符

YAML示例展示：

1）demo1

1. **---**
2. "诗仙"**:** "李白"
3. 或者
4. "诗仙"**:**
5. "李白"

2）demo2

1. #数组的例子
2. **---**
3. **-** "李白"
4. **-** "杜甫"
5. **-** "白居易"
6. **-** "唐僧"

3）demo3

1. #使用一行表示数组的例子
2. **---**
3. "诗人"**:** **[**"李白"**,**"杜甫"**,**"白居易"**]**

4)demo4

1. #键值对和数组符合例子：
2. **---**
3. "诗人"**:**
4. **-** "李白"
5. **-** "杜甫"
6. **-** "白居易"

5)demo5

1. #复杂案例
2. **---**
3. **-** "诗人"**:**
4. **-** 唐代**:**
5. **-** "李白"
6. **-** "杜甫"
7. **-** 宋代**:**
8. **-** "苏轼"
9. **-** "李清照"

6）demo6

1. #喜欢的电影
2. **---**
3. **-** 芳华
4. **-** 战狼
5. **-** 霸王别姬

7）demo7

1. #人物描述
2. **---**
3. **-** 姓名**:** 李白
4. 年龄**:** **61**
5. 作品**:** 蜀道难
6. 好友**:** 汪伦

8）demo8

1. #跨行文本（计算机理解为一行）
2. **---**
3. 自我介绍**:** **>**
4. 字太白**,**号青莲居士**,**
5. 唐代诗人**,**祖籍陇西郡**,**
6. 今甘肃省平凉市

9）demo9

1. #跨行文本（计算机理解为多行）
2. **---**
3. 自我介绍**:** **|**
4. 字太白**,**号青莲居士**,**
5. 唐代诗人**,**祖籍陇西郡**,**
6. 今甘肃省平凉市

10）demo10

注意-和:后面必须有空格。

1. #一张发票
2. **---**
3. 发票编号**:** **34843**
4. 日期**:** **2028-12-12**
5. 商品**:**
6. **-** 商品编号**:** BL394D
7. 描述**:** 足球
8. 价格**:** **100**
9. **-** 商品编号**:** BL4438H
10. 描述**:** 棒球
11. 价格**:** **200**
12. 税费**:** **10.00**
13. 总价**:** **310.00**
14. 备注**:** **>**
15. 本次采购商品均
16. 属于球类运动商品**.**

11）demo11

1. #错误日志
2. **---**
3. 时间**:** **2028-10-01** **15:01:42**
4. 用户**:** ed
5. 错误信息**:**
6. **-** 文件**:** nginx**.**conf
7. 行号**:** **23**
8. 错误编码**:** "0x3D5FF1"
9. **-** 文件**:** test**.**php
10. 行号**:** **12**
11. 错误代码**:** "0xA4C51E"
12. 警告信息**:** **|**
13. 你有两个错误信息需要查看**,**
14. 一条是配置文件错误**,**
15. 一条是脚本语法错误**,**
16. 具体内容参考错误信息**.**

Playbook语法格式要求如下：

* playbook采用YAML格式编写
* playbook文件中由一个或多个play组成
* 每个play中可以包含:
* hosts(主机)、tasks(任务)、vars(变量)等元素组成
* 使用ansible-playbook命令运行playbook剧本

**步骤一：测试Playbook语法格式**

1）编写第一个Playbook（剧本）

hosts、tasks、name是关键词（不可修改），ping是模块，调用不同模块完成不同任务。

1. **[**root@control ansible**]**# vim **~**/ansible/test**.**yml
2. **---**
3. **-** hosts**:** all                                #hosts定义要远程谁？
4. tasks**:**                                    #tasks定义远程后要执行的任务有哪些？
5. **-** name**:** This is my first playbook #name后面的具体内容可以任意
6. ping**:**
7. **[**root@control ansible**]**# ansible**-**playbook **~**/ansible/test**.**yml

执行效果如图-1所示。



图-1

2）定义多个主机和任务的剧本

hosts由一个或多个组或主机组成，逗号分隔，tasks由一个或多个任务组成，多个任务按顺序执行，执行ansible-playbook命令可以使用-f选项自定义并发量。

1. **[**root@control ansible**]**# vim **~**/ansible/test**.**yml
2. **-** hosts**:** test**,**webserver
3. tasks**:**
4. **-** name**:** This is my first playbook #name后面的内容可以任意
5. ping**:**
6. **-** name**:** Run a shell command
7. shell**:** touch **~/**shell**.**txt
8. #hosts定义需要远程哪些被管理主机，hosts是关键词
9. #tasks定义需要执行哪些任务，tasks是关键词
10. #第一个任务调用ping模块**,**该模块没有参数
11. #第二个任务调用shell模块在被管理主机创建一个空文件**~/**shell**.**txt
12. **[**root@control ansible**]**# ansible**-**playbook **~**/ansible/test**.**yml **-**f **5**
13. ## 验证：到node1、node3、node4主机分别执行命令ls **/**root**/**shell**.**txt查看是否有该文件

3）多个play的Playbook文件

1. **[**root@control ansible**]**# vim **~**/ansible/test**.**yml
2. #第一个play剧目
3. **---**
4. **-** hosts**:** test
5. tasks**:**
6. **-** name**:** This is first play
7. ping**:**
8. #第二个play剧目
9. **-** hosts**:** webserver
10. tasks**:**
11. **-** name**:** This is second play
12. ping**:**

**步骤二：Playbook应用案例**

1）用户管理，创建系统账户、账户属性、设置密码（ansible-doc user）。

1. **[**root@control ansible**]**# vim **~**/ansible/test\_john**.**yml
2. **---**
3. **-** hosts**:** webserver
4. tasks**:**
5. **-** name**:** Add the user 'johnd'
6. user**:**
7. name**:** johnd
8. uid**:** **1040**
9. group**:** daemon
10. password**:** "{{ '123' | password\_hash('sha512') }}"
11. #hosts定义需要远程的对象是webserver组，hosts是关键词
12. #tasks定义需要执行的任务，tasks是关键词
13. # name是第一个任务的描述信息，描述信息可以任意
14. # user是第一个任务需要调用的模块，user下面的缩进内容是给user模块的参数
15. # name是需要创建的用户名，uid是用户ID号
16. # group是用户属于哪个基本组
17. # password是用户的密码，密码是**123**，密码经过sha512算法加密
18. **[**root@control ansible**]**# vim **~**/ansible/user\_james**.**yml
19. **---**
20. **-** hosts**:** webserver
21. tasks**:**
22. **-** name**:** Add 'james' **with** a bash shell
23. user**:**
24. name**:** james
25. shell**:** /bin/bash
26. groups**:** bin**,**adm
27. password**:** "{{ '123' | password\_hash('sha512') }}"
28. #与上一个案例类似，groups指定用户属于哪些附加组**.**
29. **[**root@control ansible**]**# vim **~**/ansible/user\_johnd**.**yml
30. **---**
31. **-** hosts**:** webserver
32. tasks**:**
33. **-** name**:** Remove the user 'johnd'
34. user**:**
35. name**:** johnd
36. state**:** absent
37. #删除系统账户johnd，state的值设置为absent是删除用户

2）使用playbook管理逻辑卷

准备工作：给node2主机再添加一块磁盘（以下实验磁盘名称仅为参考，不要照抄）。

（ansible-doc parted，ansible-doc lvg，ansible-doc lvol）

1. **[**root@control ansible**]**# vim **~**/ansible/lvm**.**yml
2. **---**
3. **-** hosts**:** node2                            #远程node2主机
4. tasks**:**
5. **-** name**:** Create a **new** primary partition **with** a size of 1GiB #任务的描述信息
6. parted**:** #调用parted模块进行分区
7. device**:** /dev/sdb #对/dev/sdb磁盘进行分区**(**磁盘名称不要照抄**)**
8. label**:** gpt #分区表类型为gpt，或msdos
9. number**:** **1** #分区编号**(**创建第几个分区**)**
10. state**:** present #present是创建分区**,**absent是删除分区
11. part\_start**:** 1MiB #分区的开始位置（默认从最开始位置分区）
12. part\_end**:** 1GiB #分区的结束位置（不写就分到磁盘最后位置）
13. **-** name**:** Create a volume group on top of **/**dev**/**sdb1 #第二个任务的描述信息
14. lvg**:** #调用lvg模块**,**创建VG卷组
15. vg**:** my\_vg #要创建的卷组名称
16. pvs**:** /dev/sdb1 #使用哪个分区创建PV
17. **-** name**:** Create a logical volume of 512m #第三个任务的描述信息
18. lvol**:** #调用lvol模块创建LV
19. vg**:** my\_vg #使用哪个VG创建LV
20. lv**:** my\_lv #需要创建的LV名称
21. size**:** 512m #要创建的LV大小**,**可以不指定单位，默认单位m

3）使用playbook管理软件（ansible-doc yum）

RHEL或CentOS系统中的软件有组包的概念，使用yum grouplist或者dnf grouplist可以查看组包的名称。

1. **[**root@control ansible**]**# vim **~**/ansible/package**.**yml
2. **---**
3. **-** hosts**:** webserver #需要远程的主机是谁
4. tasks**:** #定义剧本需要执行的任务
5. **-** name**:** Install a list of packages #第一个任务的描述信息
6. yum**:** #调用yum模块安装软件
7. name**:** #安装软件的名字，它的值有多个，使用数组**-**
8. **-** httpd #安装httpd软件
9. **-** mariadb #安装mariadb软件
10. **-** mariadb**-**server #安装mariadb**-**server
11. **-** name**:** install the 'RPM Development Tools' package group #第二个任务的描述信息
12. yum**:** #调用yum模块安装软件组包
13. name**:** "@RPM Development Tools" #安装哪个组包，@是关键词
14. **-** name**:** update software #第三个任务的描述信息
15. yum**:** #调用yum模块升级软件
16. name**:** '\*' #需要升级哪些软件
17. state**:** latest #latest代表升级软件
18. #备注**:**state的值可以是**(**present**|**absent**|**latest**)**
19. #present代表安装软件**(**默认是present**)**；absent代表卸载软件
20. #latest代表升级软件

**4 案例4：Playbook应用案例**

**4.1 问题**

沿用练习三，继续练习Ansible 特殊模块并掌握自定义变量的方式，具体要求如下：

* 熟悉setup与debug模块
* 熟悉各种常见的变量定义方式

**步骤一：Ansible特殊模块**

1）setup模块

ansible\_facts用于采集被管理设备的系统信息，所有收集的信息都被保存在变量中，每次执行playbook默认第一个任务就是Gathering Facts，使用setup模块可以查看收集到的facts信息。

1. **[**root@control ansible**]**# ansible test **-**m setup
2. **192.168.4.10** **|** SUCCESS **=>** **{**
3. "ansible\_facts"**:** **{**
4. "ansible\_all\_ipv4\_addresses"**:** **[**
5. … 省略部分内容…

试试自己找出下列变量：

* ansible\_all\_ipv4\_addresses #IP地址
* ansible\_bios\_version #主板BIOS版本
* ansible\_memtotal\_mb #总内存
* ansible\_hostname #主机名
* ansible\_fqdn #主机的域名
* ansible\_devices.sda.partitions.sda1.size #某个磁盘分区的大小

2）debug模块

debug模块可以显示变量的值，可以辅助排错，通过msg可以显示变量的值，变量需要使用{{}}扩起来。

1. **[**root@control ansible**]**# vim **~**/ansible/debug**.**yml
2. **---**
3. **-** hosts**:** test
4. tasks**:**
5. **-** debug**:**
6. msg**:** "主机名是:{{ ansible\_hostname }}"
7. **-** debug**:**
8. msg**:** "总内存大小:{{ ansible\_memtotal\_mb }}"
9. #备注调用debug模块显示某些具体的变量值
10. #debug模块可以显示变量的值，可以辅助排错

**步骤二：定义变量的方法**

Ansible支持十几种定义变量的方式，这里我们仅介绍其中一部分变量。

下面是根据优先级排序的定义方式：

1. Inventory变量
2. Host Facts变量
3. Playbook变量
4. Playbook提示变量
5. 变量文件
6. 命令行变量

1）Inventory变量(在主机清单配置文件中定义变量）。

1. **[**root@control ansible**]**# vim **~**/ansible/inventory
2. **[**test**]**
3. node1 iname**=**"nb"
4. **[**proxy**]**
5. node2
6. **[**webserver**]**
7. node**[3:4]**
8. **[**webserver**:**vars**]**
9. iname**=**"dachui"
10. #备注，在node1主机后面给该主机添加变量iname**,**值为nb**.**
11. #给webserver组定义变量**,**vars是关键词不可以改变**,**webserver是上面定义的组
12. #给这个组定义变量iname**=**"dachui"

下面编写剧本调用刚才的变量：(在剧本中需要调用变量是要使用{{}})

1. **[**root@control ansible**]**# vim **~**/ansible/inventory\_var**.**yml
2. **---**
3. **-** hosts**:** node1**,**webserver #定义需要远程管理的主机是谁
4. tasks**:** #剧目要完成哪些任务
5. **-** name**:** create a user **with** **var.** #剧目中的第一个任务描述信息
6. user**:** #调用user模块创建用户
7. name**:** "{{ iname }}" #需要创建的用户名是iname这个变量
8. #注意事项：
9. #在ansible剧本中当调用变量时，开始位置就调用变量**,**就需要在**{{}}**外面加双引号
10. #如果是在后面或者中间位置调用变量**{{}}**外面可以不加双引号
11. #如**:**
12. # "{{ iname }}"
13. # nihao **{{** iname **}}**

2）Host Facts变量（可以直接调用ansible收集的系统信息）

1. **[**root@control ansible**]**# vim **~**/ansible/facts\_var**.**yml
2. **---**
3. **-** hosts**:** test
4. tasks**:**
5. **-** name**:** create user**.**
6. user**:**
7. name**:** "{{ansible\_hostname}}"
8. #定义剧本，远程所有被管理主机，调用user模块，创建用户
9. #需要创建的用户名ansible\_hostname是一个ansible\_facts变量
10. #验证： 到node1主机查看是否有一个与主机名同名的用户

3）Playbook变量(使用vars关键词可以在playbook内定义变量）。

1. **[**root@control ansible**]**# vim **~**/ansible/playbook\_var**.**yml
2. **---**
3. **-** hosts**:** test
4. vars**:** #vars是关键词，用来定义变量用的
5. iname**:** heal #具体变量名是iname，值是heal
6. ipass**:** '123456' #再定义一个变量名是ipass，值是**123456**
7. #注意密码必须是字符串，需要引号
8. tasks**:** #tasks定义需要执行的任务
9. **-** name**:** Use variables create user**.** #给任务写个描述信息
10. user**:** #调用user模块创建用户
11. name**:** "{{ iname }}" #用户名的是前面定义的变量
12. password**:** "{{ ipass | password\_hash('sha512') }}"
13. #密码是前面定义好的ipass**,**管道给password\_hash把密码加密**.**

4）Playbook提示变量（根据提示输入变量的值）。

1. **[**root@control ansible**]**# vim **~**/ansible/prompt\_var**.**yml
2. **---**
3. **-** hosts**:** test
4. vars\_prompt**:** #vars\_prompt是关键词，定义提示变量
5. **-** name**:** iname #通过name定义变量名为iname
6. prompt**:** "请输入用户名" #执行剧本时提示信息
7. **private:** no #是否是隐私数据**,**no代表不是（屏幕可以显示）
8. **-** name**:** ipasswd #通过name再定义变量，变量名为ipasswd
9. prompt**:** "请输入密码" #执行剧本时提示信息
10. **private:** yes #是否是隐私数据，yes代表是**(**隐私数据屏幕不显示**)**
11. tasks**:** #定义剧本需要执行的任务
12. **-** name**:** Create a user**.** #定义第一个任务的描述信息
13. user**:** #调用user模块创建用户，设置用户密码
14. name**:** "{{ iname }}" #用户名是前面创建的变量iname
15. password**:** "{{ ipasswd | password\_hash('sha512') }}"
16. #密码是前面定义的变量ipasswd，密码管道给password\_hash加密**.**

5）单独定义个变量文件，在playbook中用vars\_files调用该文件。

1. **[**root@control ansible**]**# vim **~**/ansible/file\_var**.**yml
2. **---**
3. **-** hosts**:** test
4. vars\_files**:** variables**.**yml #当变量比较多时，专门定义一个文件用来存变量
5. tasks**:**
6. **-** name**:** create user**.**
7. user**:**
8. name**:** "{{ iname }}"
9. password**:** "{{ ipass | password\_hash('sha512') }}"
10. #调用user模块创建用户
11. #用户名是变量文件variables**.**yml中定义的变量iname，密码也是变量文件中定义的变量
12. **[**root@control ansible**]**# vim **~**/ansible/variables**.**yml
13. **---**
14. iname**:** cloud
15. ipass**:** '123456'

6）执行ansible-playbook命令时使用-e参数定义变量

1. **[**root@control ansible**]**# vim **~**/ansible/command\_var**.**yml
2. **---**
3. **-** hosts**:** test
4. tasks**:**
5. **-** name**:** create user
6. user**:**
7. name**:** "{{ iname }}"
8. password**:** "{{ ipass | password\_hash('sha512') }}"
9. **[**root@control ansible**]**# ansible**-**playbook command\_var**.**yml **\**
10. **-**e iname**=**"beth" **-**e ipass**=**"123456"

附加思维导图，如图-2所示：

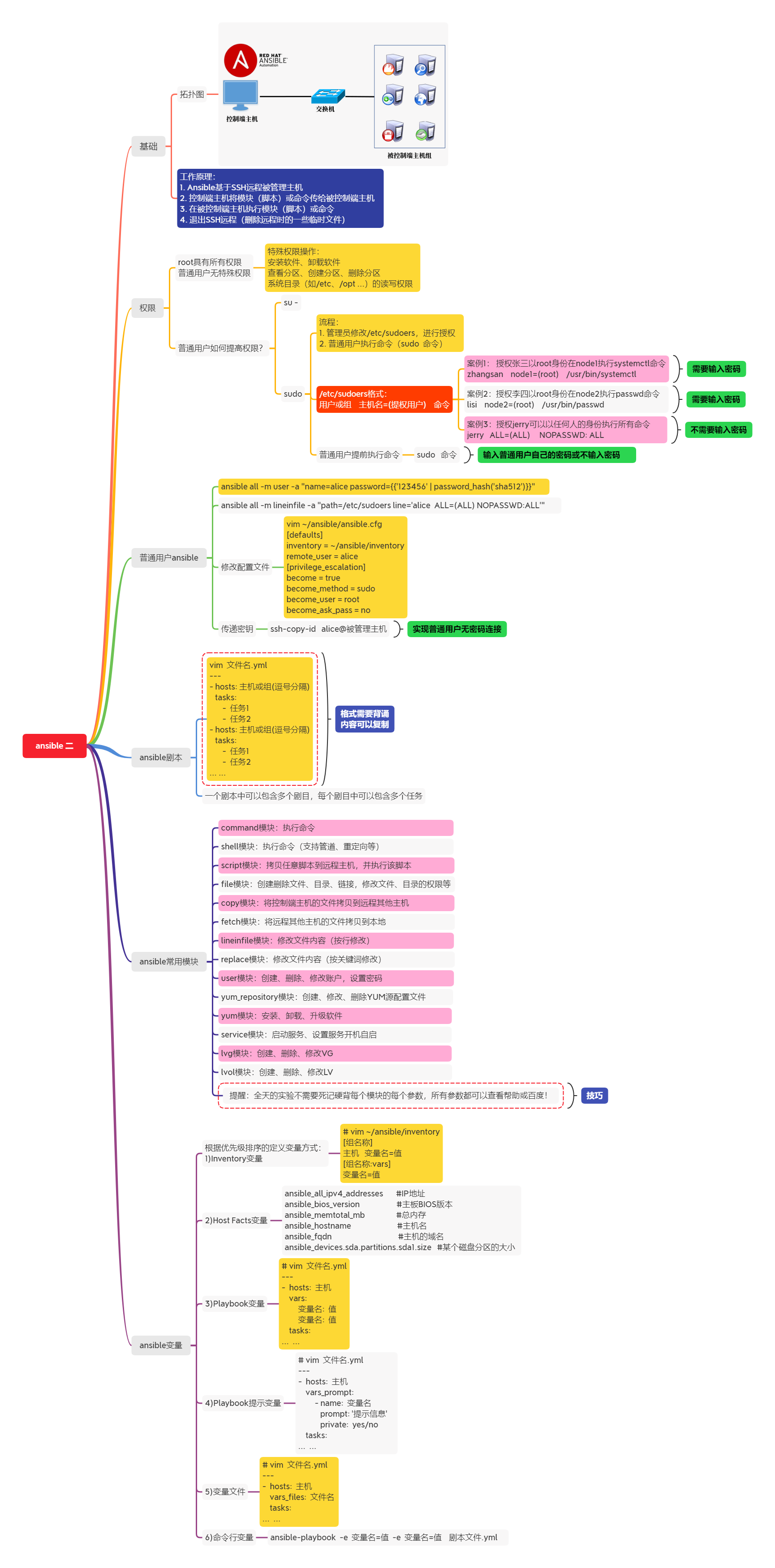
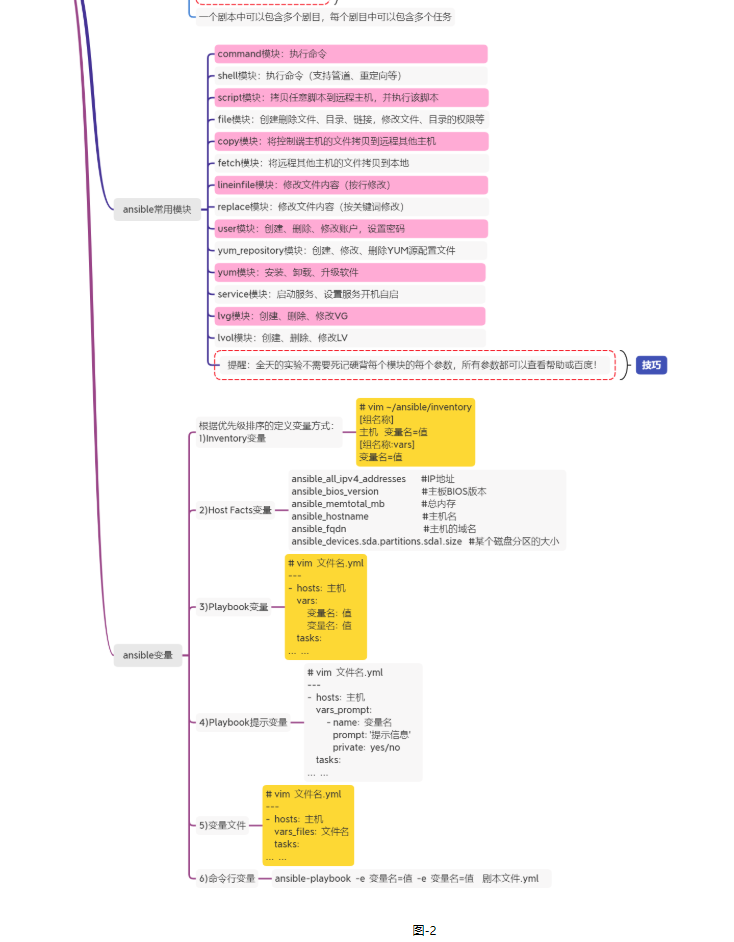
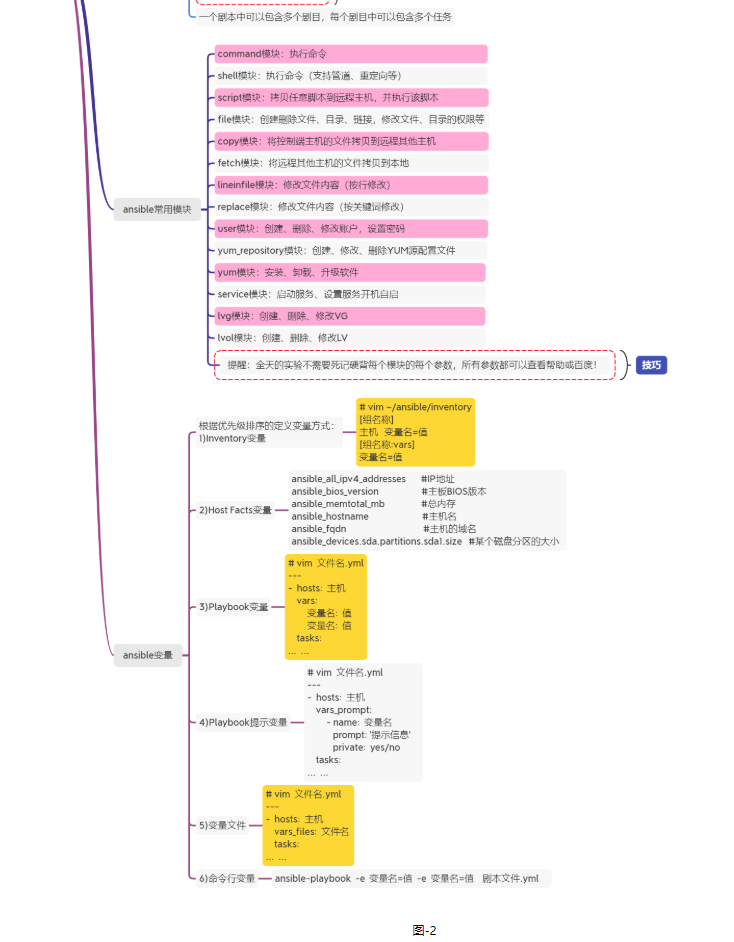


图-2