A+

ansible笔记（1）：ansible的基本概念

一些基础概念

ansible是什么?

它是一个"配置管理工具"，它是一个"自动化运维工具"，如果你没有使用过任何配置管理工具，不要害怕，看完这篇文章，你自然会对ansible有所了解。

ansible能做什么?

正如其他配置管理工具一样，ansible可以帮助我们完成一些批量任务，或者完成一些需要经常重复的工作。

比如：同时在100台服务器上安装nginx服务，并在安装后启动它们。

比如：将某个文件一次性拷贝到100台服务器上。

比如：每当有新服务器加入工作环境时，你都要为新服务器部署redis服务，也就是说你需要经常重复的完成相同的工作。

这些场景中我们都可以使用到ansible。

看到这里，你可能会说，我编写一些脚本，也能够满足上面的工作场景，为什么还要使用ansible呢？没错，使用脚本也可以完成这些工作，不过我还是推荐你使用a
nsible支持一些优秀的特性，比如"幂等性"，"幂等性"是什么意思呢？举个例子，你想把一个文件拷贝到目标主机的某个目录上，但是你不确定此目录中是否已经存
你使用ansible完成这项任务时，就非常简单了，因为如果目标主机的对应目录中已经存在此文件，那么ansible则不会进行任何操作，如果目标主机的对应目录中并
件，ansible就会将文件拷贝到对应目录中，说白了，ansible是"以结果为导向的"，我们指定了一个"目标状态"，ansible会自动判断，"当前状态"是否与"目标状态"
致，则不进行任何操作，如果不一致，那么就将"当前状态"变成"目标状态"，这就是"幂等性"，"幂等性"可以保证我们重复的执行同一项操作时，得到的结果是一样
在很多场景中相对于脚本来说都有一定优势，单单这样说，可能并不容易理解，当你在后面真正使用到时，自然会有自己的体会，所以此处不用纠结，继续向下看。

如果你了解过其他的配置管理工具，比如puppet或者saltstack，那么你一定知道，如果我们想要使用puppet管理100台主机，就要在这100台主机上安装puppet对
（客户端代理程序），而ansible则不同，ansible只需要依赖ssh即可正常工作，不用在受管主机上安装agent，也就是说，只要你能通过ssh连接到对应主机，你就
e管理对应的主机。

经过上述描述，我想你应该对ansible已经有了一个初步的、大概的印象：

ansible是一个配置管理工具，可以帮助我们完成一些批量工作或者重复性工作，ansible通过ssh管理其他受管主机，并且具有一些特性，比如幂等性、剧本、模板，
会慢慢的介绍这些特性以及怎样使用ansible。

怎样使用ansible呢？我们通过一条简单的命令开始认识它吧，命令如下

注：执行如下命令前，需要进行一些配置，如下命令才能正常执行，后文中会对这些操作进行描述，此处先行略过

```
1 ansible 10.1.1.60 -m ping
```

上述命令表示，使用ansible去ping 10.1.1.60这台主机，很容易理解吧。

"ping"是ansible中的一个模块，这个模块的作用就是ping对应的主机，ansible调用ping模块，就相当于我们手动执行ping命令一样，上述命令中的"-m ping"表示
块，当然，ansible肯定不止这一个模块，它有很多模块，不同的模块可以帮助我们完成不同的工作，你应该已经猜到了，我们在实际使用时，会使用到各种模块，a
这些模块完成实际任务的。

刚才，我们使用了一个简单的ansible命令作为示例，但是如果想要让上述命令正常执行，则必须同时满足两个最基本的条件，如下

条件一、ansible所在的主机可以通过ssh连接到受管主机。

条件二、受管主机的IP地址等信息已经添加到ansible的"管理清单"中。

之前说过，ansible不用在受管主机上安装agent，但是它需要依赖ssh，所以，条件一并不难理解，但是，在满足条件一的情况下，还要同时满足条件二，也就是说
所在的主机能够通过ssh连接到受管主机，仍然需要将受管主机的IP地址、ssh端口号等信息添加到一个被称作为"清单(Inventory)"的配置文件中，如果对应的主机信
的"清单"中不存在，那么ansible则无法操作对应主机，后文会详细的介绍怎样配置ansible的"清单"。

好了，基本概念先了解到这里，现在需要动动手了。

一些基础配置

我们首先要做的就是安装ansible。

但是在安装之前，先介绍一下我的演示环境。

我有四台主机，IP地址分别如下

10.1.1.71

10.1.1.70

10.1.1.61

10.1.1.60

我将主机10.1.1.71（后文中简称71）作为配置管理主机，所以我们需要在71上安装ansible，剩下的主机作为受管主机，主机71和主机70的的操作系统版本为centc
和主机60的操作系统版本为centos6.9。

我使用yum源的方式安装ansible，因为安装ansible需要epel源，所以我配置了阿里的epel源和centos7系统镜像源，yum源配置如下

```
# pwd
/etc/yum.repos.d

# cat aliBase.repo
[aliBase]
name=aliBase
baseurl=https://mirrors.aliyun.com/centos/$releasever/os/$basearch/
enabled=1
gpgcheck=1
gpgkey=https://mirrors.aliyun.com/centos/$releasever/os/$basearch/RPM-GPG-KEY-CentOS-$releasever

# cat aliEpel.repo
[aliEpel]
name=aliEpel
baseurl=https://mirrors.aliyun.com/epel/$releasever\Server/$basearch/
enabled=1
gpgcheck=0
```

yum源配置完成后，安装ansible

```
yum install ansible
```

此时yum源中对应的版本为ansible-2.4.2.0-1

安装完毕，不过别急，我们还需要做一些其他的基本配置，在介绍ansible的概念时，我们说过，如果想要通过ansible管理某主机，还需要将对应主机的信息添加到
置清单"中，清单中没有的主机无法通过ansible进行配置管理，现在，我们就来介绍一下ansible的"清单"，当安装完ansible以后，ansible会提供一个默认的"清单"
是/etc/ansible/hosts，打开此文件，你会看到一些配置示例，没错，还是熟悉的配方，还是熟悉的味道，此文件使用的就是INI的配置风格，那么，我们一起来看看
吧。

以我们的演示环境为例，我们想要通过ansible主机管理60主机，所以，最直接的方式就是将它的IP地址写入到/etc/ansible/hosts文件中，配置如下，在/etc/ansib
底部写入如下IP

10.1.1.60

就是这么简单，那么，完成上述配置，就能够通过ansible主机管理10.1.1.60这台主机了吗？我们来动手试试，看看会发生什么情况。

执行之前的示例命令：ansible 10.1.1.60 -m ping

使用ansible去ping主机10.1.1.60，返回结果如下

```
# ansible 10.1.1.60 -m ping
10.1.1.60 | UNREACHABLE! => {
    "changed": false,
    "msg": "Failed to connect to the host via ssh: Permission
 denied (publickey,gssapi-keyex,gssapi-with-mic,password).\r\
n",
    "unreachable": true
}
```


从命令的返回信息中可以看到，10.1.1.60不可达，也就是说，ansible无法通过ssh连接到主机60。

返回上述信息是正常的，因为ansible主机并不知道10.1.1.60这台主机的用户名和密码，所以ansible无法通过ssh连接到它。

所以，我们还需要在清单中，配置10.1.1.60主机的ssh信息，才能够进行正确的进行连接，配置示例如下：

```
10.1.1.60 ansible_port=22 ansible_user=root ansible_ssh_pass=123123
```

修改清单文件，在之前的主机IP后加入ssh的相关配置信息，如上图所示

ansible_port 用于配置对应主机上的sshd服务端口号，在实际的生产环境中，各个主机的端口号通常不会使用默认的22号端口，所以用此参数指定对应端口。

ansible_user 用于配置连接到对应主机时所使用的用户名称。

ansible_ssh_pass 用于配置对应用户的连接密码。

所以，上图中的配置表示，10.1.1.60这台主机的sshd服务监听在22号端口，当ansible通过ssh连接到主机60时，会使用主机60的root用户进行连接，主机60的roo
123123

好了，主机60的ssh信息已经配置完毕，我们再来尝试一下，看看之前的命令能不能正常执行，如下

```
# ansible 10.1.1.60 -m ping
10.1.1.60 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
#
```

可以看到，上述命令已经正常执行了，ansible主机成功的ping通了10.1.1.60，从此以后，我们就可以通过ansible主机，管理10.1.1.60这台主机了。

其实，为了更加方便的使用，ansible还支持对主机添加别名，当主机存在别名时，我们可以通过主机的"别名"管理对应主机。

比如，我们想要将10.1.1.60这台主机的别名命名为test60，那么，我们在配置清单时，可以进行如下配置

如上图所示，当为主机配置别名时，主机的IP地址必须使用anible_host关键字进行指明，否则ansible将无法正确的识别对应的主机。

主机的别名配置完成后，则可以使用主机的别名管理对应主机，示例如下。

不过，如果你只使用了上述方式配置了主机，则无法通过主机的IP进行管理了，除非你同时使用了别名的方式与IP的方式配置两个主机条目。

注意：上述配置参数都是ansible2.0版本以后的写法，2.0版本之前，应遵从如下写法

ansible_port应该写成ansible_ssh_port

ansible_user应该写成ansible_ssh_user

ansible_host应该写成ansible_ssh_host

因为当前演示环境的ansible版本为2.4，所以，我们使用新的写法进行演示，2.4版本同时也兼容之前的语法。

上述参数，其实都是为了创建ssh连接所使用的，而说到ssh，我们都知道，创建ssh连接时，可以基于密码进行认证，也可以基于密钥进行认证，而在生产环境中，
性，我们通常会基于密钥进行ssh认证，甚至会禁用密码认证，那么，当ansible主机需要与受管主机建立ssh连接时，能够基于密钥进行认证码？必须能的。

其实，在实际的使用环境中，我们通常会在"配置管理机（ansible主机）"中生成密钥，然后通过公钥认证的方式连接到对应的受管主机中，如果你对基于密钥认证的
别了解，则可以参考如下文章，此处不再对相应配置进行详细的描述：

http://www.zsythink.net/archives/2375

那么，我们就在ansible主机中生成密钥，并进行相应的配置吧。

首先，生成默认格式的密钥对，私钥与公钥。

```
1 # ssh-keygen
```

然后将生成的公钥加入到10.1.1.60的认证列表

```
1 # ssh-copy-id -i /root/.ssh/id_rsa.pub root@10.1.1.60
```

好了，公钥认证的相关操作配置完成，此刻，我们已经可以通过ansible主机免密码连接到主机60中了。

因为配置了密钥认证，所以可以实现免密码创建ssh连接，既然已经能够免密码创建ssh连接，那么在配置"主机清单"时，就没有必要再提供对应主机的用户名与密码
完成了密钥认证的相关配置后，我们可以将清单中的配置精简为如下格式。

或者使用别名的格式

当然，如果你的受管服务器中的sshd服务使用了默认的22号端口，上述配置中的ansible_port也是可以省略的，为了方便演示，演示环境中的所有受管主机均使用默
号。

如果你的ansible主机上同时存在多对密钥，有可能需要通过不同的密钥连接不同的受管主机，这个时候，你可以通过ssh-agent帮助我们管理密钥，如果你还不了解
那么可以参考如下文章：

http://www.zsythink.net/archives/2407

如果你不想使用ssh-agent管理密钥，也可以通过ansible_ssh_private_key_file参数，指定连接对应主机时所使用的私钥，由于演示环境中并没有同时使用多对密钥再赘述。

在今后的演示中，默认使用密钥认证的方式连接到对应主机，我会提前配置好各个受管主机的密钥认证，后文中将不再对密钥认证的配置过程进行描述。

好了，说了这么多，我想你应该已经了解了ansible的基本概念，以及ansible的一些最基本的配置，在之后的文章中，我们会徐徐渐进，慢慢的介绍ansible的。

下一篇ansible文章直达链接：ansible清单配置详解

我的微信公众号

关注"实用运维笔记"微信公众号，当博客中有新文章时，可第一时间得知哦~