

CentOS 7 中安装并使用自动化运维工具

Ansible

一、简介

Ansible 是一款为类 Unix 系统开发的自由开源的配置和自动化工具，它用 Python 写成，糅合了批量操作系统配置、批量程序的部署、批量运行命令等功能。但是有一个不同的优点是我们在不需要在节点中安装任何客户端。它使用 SSH 来和节点进行通信。

二、Ansible 特性

基于 Python 语言实现，由 Paramiko, PyYAML 和 Jinja2 三个关键模块；

ansible 是工作在 agentless 模式下具有幂等性。ansible 在控制端只需要告诉监控端的期望状态就可以实现批量部署。

默认使用 SSH 协议；

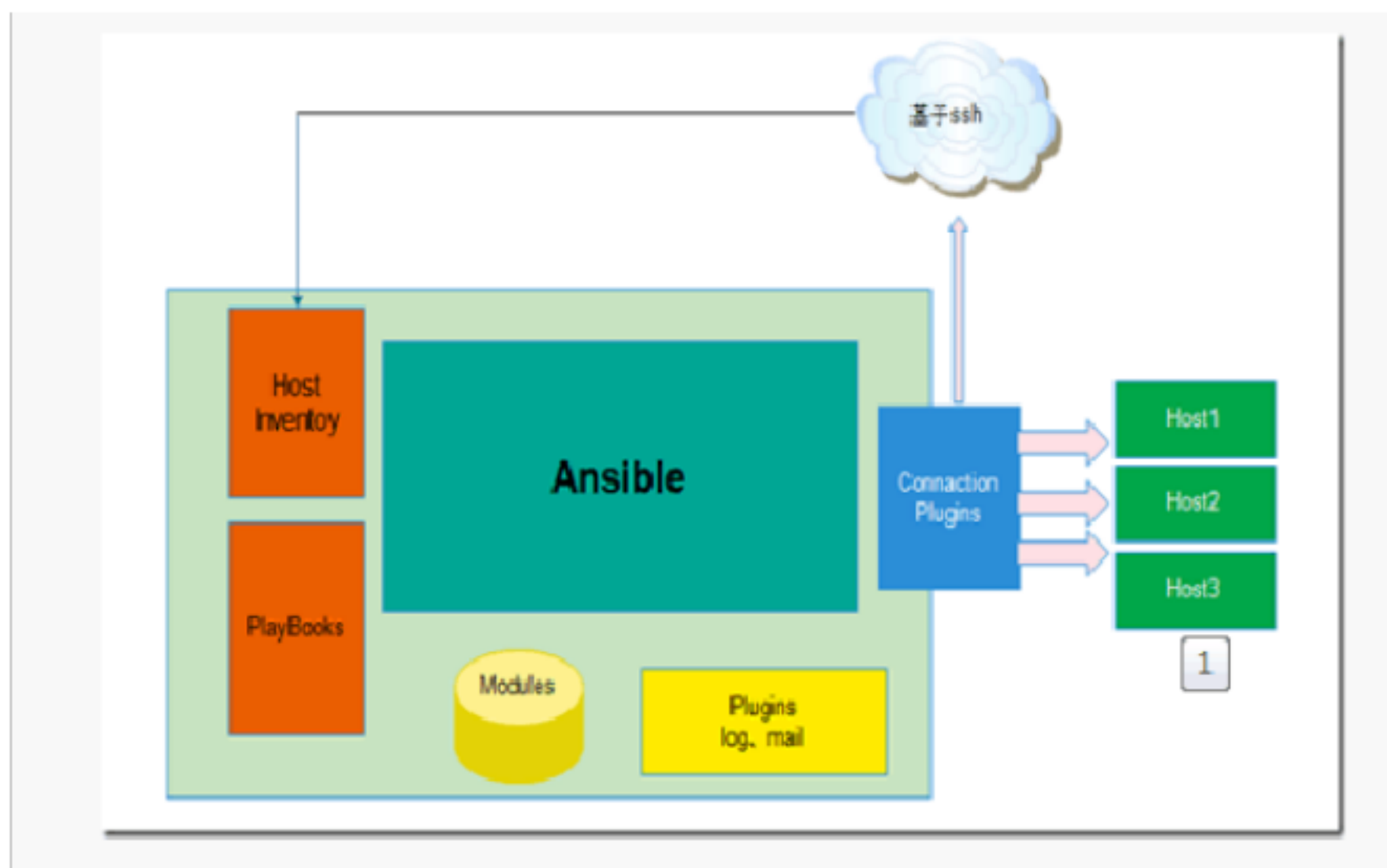
- (1) 基于密钥认证；
- (2) 在 inventory 文件中指定账号和密码；

基于“模块”完成各种“任务”

支持自定义模块：支持各种编程语言

可以使用 YAML 语言定制 playbook

三、框架介绍



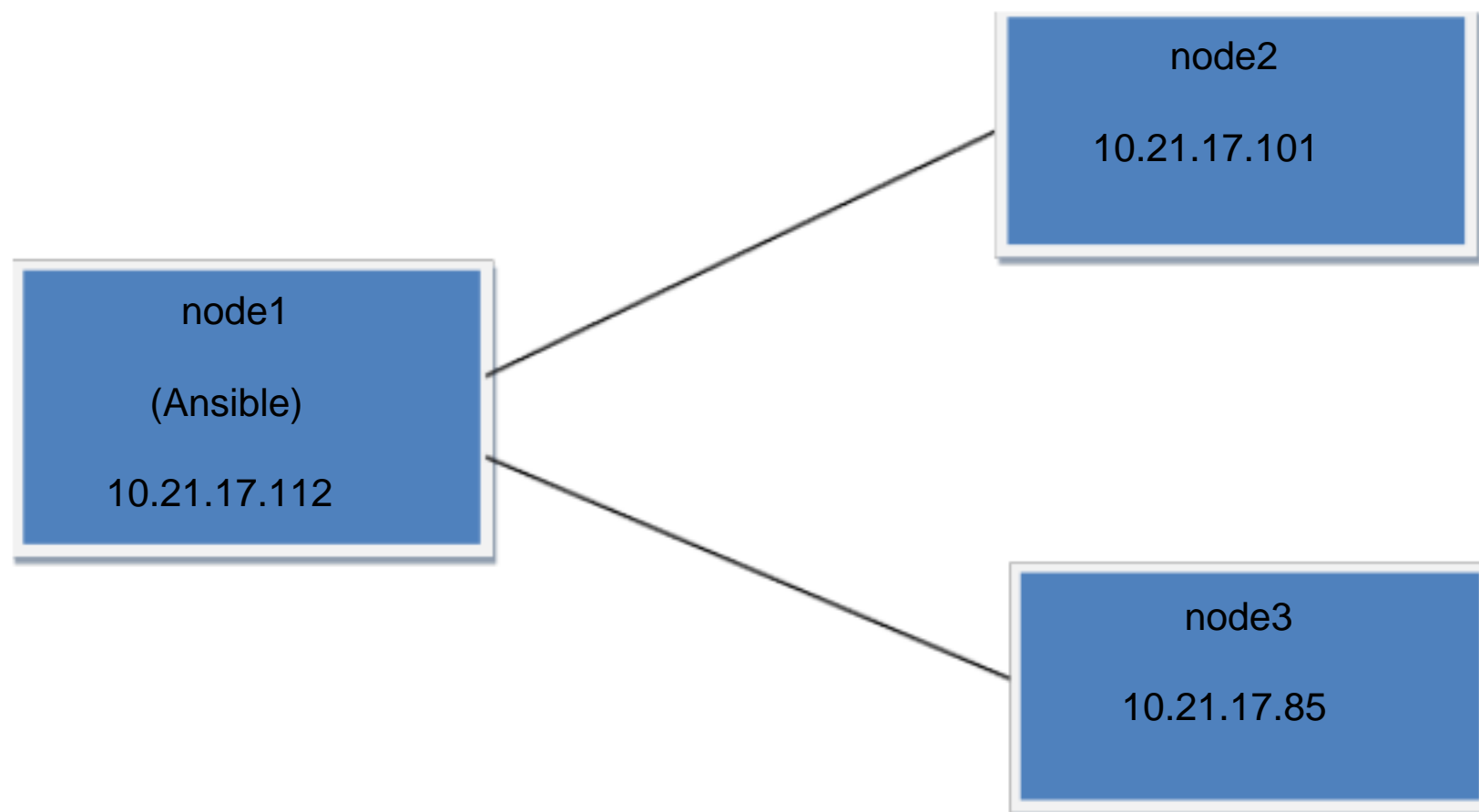
四、Ansible 核心组件说明：

ansible 是基于模块工作的 ansible 本身没有批量部署的能力。真正具有批量部署的是 ansible 所运行的模块 ansible 只是提供一种框架。架构包括

- 连接插件 connection plugins 负责和被监控端实现通信。
- Host Inventory: 指定操作的主机，是一个配置文件里面定义监控的主机

- 各种模块核心模块 command 模块自定义模块
- 借助于插件完成记录日志邮件等功能
- PlayBooks: 剧本执行多个任务时。并非必需可以让节点一次性运行多个任务

五、环境拓扑图



六、安装 ansible

第一步：设置 EPEL 仓库

Ansible 仓库默认不在 yum 仓库中，因此我们需要使用下面的命令启用 epel 仓库。

```
[root@localhost~]# rpm -iUvh http://dl.Fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
```

第二步：使用 yum 安装 Ansible

```
[root@localhost ~]#yum install ansible
```

安装完成后，检查 ansible 版本

```
[root@localhost ~]# ansible --version
```

```
[root@localhost ~]# ansible --version
ansible 2.2.1.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = Default w/o overrides
[root@localhost ~]# █
```

第三步：设置用于节点鉴权的 SSH 密钥

```
[root@localhost ~]# ssh-keygen
```

```
[root@localhost ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
da:c1:64:d8:64:d4:50:9b:0e:14:a5:bb:20:78:23:a0 root@localhost
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           .B*O      |
|          * ..O     |
|         . * O      |
|        .. . + +    |
|       E O + . S .   |
|          O O + O    |
|           . O      |
+-----+

```

使用 ssh-copy-id 命令来复制 Ansible 公钥到节点中。

```
ssh-copy-id -i root@10.21.17.101
```

```
ssh-copy-id -i root@10.21.17.85
```

第四步：为 Ansible 定义节点的清单

文件 /etc/ansible/hosts 维护着 Ansible 中服务器的清单。

```
[root@localhost ~]#vi/etc/ansible/hosts
```

```
[test-servers]
```

```
10.21.17.85
```

```
10.21.17.101
```

保存并退出文件。

备注：如果没有设置基于 ssh 的密钥认证，可以在 hosts 文件中直接指定帐号和密码，方法如下所示：

```
[test-servers]
```

```
10.21.17.101 ansible_ssh_user=root ansible_ssh_pass=password
```

第五步：尝试在 Ansible 服务端运行命令

使用 ping 检查 'testservers 或者 ansible 节点的连通性。

```
[root@localhost ~]# ansible -m ping 'test-servers'
```

```
[root@localhost ~]# ansible -m ping 10.21.17.85
```

七、ansible 示例

1、command模块

命令模块，命令模块，默认模块，用于在远程主机执行命令

缺点：运行的命令中无法使用变量，管道。

例子 1，检查 Ansible 节点的日期

```
[root@ansible ~]# ansible -m command -a "date" 'test-servers'
```

```
[root@localhost ~]# ansible -m command -a "date" 'test-servers'  
Enter passphrase for key '/root/.ssh/id_rsa':  
10.21.17.101 | SUCCESS | rc=0 >>  
2017年 04月 06日 星期四 14:53:57 CST  
Enter passphrase for key '/root/.ssh/id_rsa':  
10.21.17.85 | SUCCESS | rc=0 >>  
2017年 04月 06日 星期四 14:53:42 CST
```

例子 2：检查 Ansible 节点的运行时间（ uptime ）

```
[root@ansible ~]# ansible -m command -a "uptime" 'test-servers'
```

例子 3：检查节点的内核版本

```
[root@ansible ~]# ansible test-servers -m command -a "uname -r"
```

例子 4：给节点增加用户

```
[root@ansible ~]# ansible 10.21.17.101 -m command -a "useradd user1"
```

例子 4：给节点删除用户

```
[root@ansible ~]# ansible 10.21.17.101 -m command -a "userdel user1"
```

2、Shell 模块

执行的命令中有管道或者变量，就需要使用 `shell`。

例子 1：给节点增加用户后，修改密码

```
[root@localhost ~]# ansible 10.21.17.101 -m shell -a 'echo user1|passwd --stdin user1'
```

3、yum , service 模块

安装软件和启动服务。

例子 1 : 给节点安装 tomcat 应用

```
[root@localhost ~]# ansible 10.21.17.101 -m yum -a name=tomcat state=present
```

例子 2 , 启动 tomcat

```
[root@localhost ~]# ansible 10.21.17.101 -m service -a name=tomcat state=started enabled=yes
```

4、raw 模块

支持管道的模块，类似 shell 模块。

例子 1 : 检查 eth0 网口

```
[root@localhost ~]# ansible 10.21.17.101 -m raw -a ifconfig|grep eth0
```