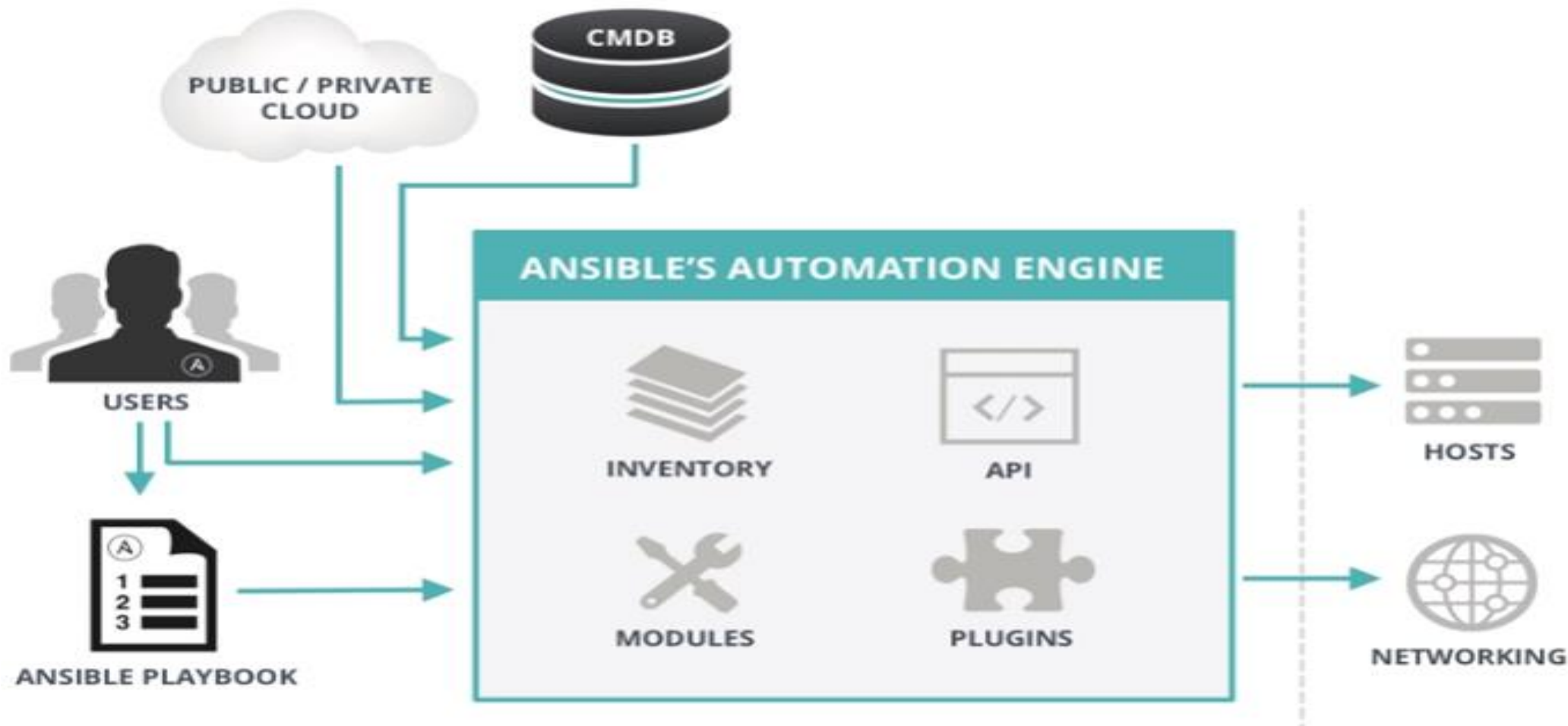




# Ansible Develop Guide

# Ansible 架构



# Ansible Develop

- 动态 Inventory
- Ansible Module
- Ansible Python API
- Ansible Plugin

# 动态Inventory

- 为什么要用动态Inventory

- 服务器信息存放在别的软件系统中

例如CMDB, Openstack, Zabbix, cobbler装机系统等

- 动态Inventory的格式

如右图所示

- 如何生成动态Inventory

- 演示如何从CMDB中获取Inventory信息(mysql, sqlalchemy)

- 如何使用Dynamic Inventory

```
{
  "webserver": {
    "hosts": [
      "192.168.33.11"
    ],
    "vars": {
      "ansible_become_user": "vagrant",
      "ansible_become_pass": "vagrant"
    }
  },
  "dbserver": {
    "hosts": [
      "192.168.33.12"
    ],
    "vars": {
      "ansible_become_user": "root",
      "ansible_become_pass": "secret"
    }
  },
  "_meta": {
    "hostvars": {
      "192.168.33.11": {
        "software": "jboss"
      },
      "192.168.33.12": {
        "software": "mysql"
      }
    }
  }
}
```

# Ansible Module

- 为什么要(是否有必要)开发自己的Ansible Module
  - 使用`ansible-doc -list` 可以列出有哪些模块可以使用
  - 如果有特殊的需求可以使用script模块
- 使用script模块来执行特殊的需求
  - 演示如何使用script模块
  - Script模块有哪些不足
    - 希望可以获取返回值
    - 需要对返回值进行判断处理
    - 需要比较复杂的参数，例如传入json字符串，对字符串进行分析
- 怎样开发自己的Ansible Module
  - 任何语言都可以编写Ansible Module，只需要返回的格式为json字符串
  - 使用shell 开发自己的Module
  - 使用python 开发自己的Module，墙裂推荐(AnsibleModule类)



# Ansible Python API

- 什么是Ansible Python API

Ansible Python API是一组用python编写的类和函数，便于使用python程序调用Ansible的核心功能

- 为什么要使用Ansible Python API

- 前一次的执行结果作为后一次任务的参数输入
- 对任务的执行结果进行定制化输出或者存储（引入callback plugin）
- 方便其他程序调用ansible的核心功能

- Ansible Python API 的调用流程

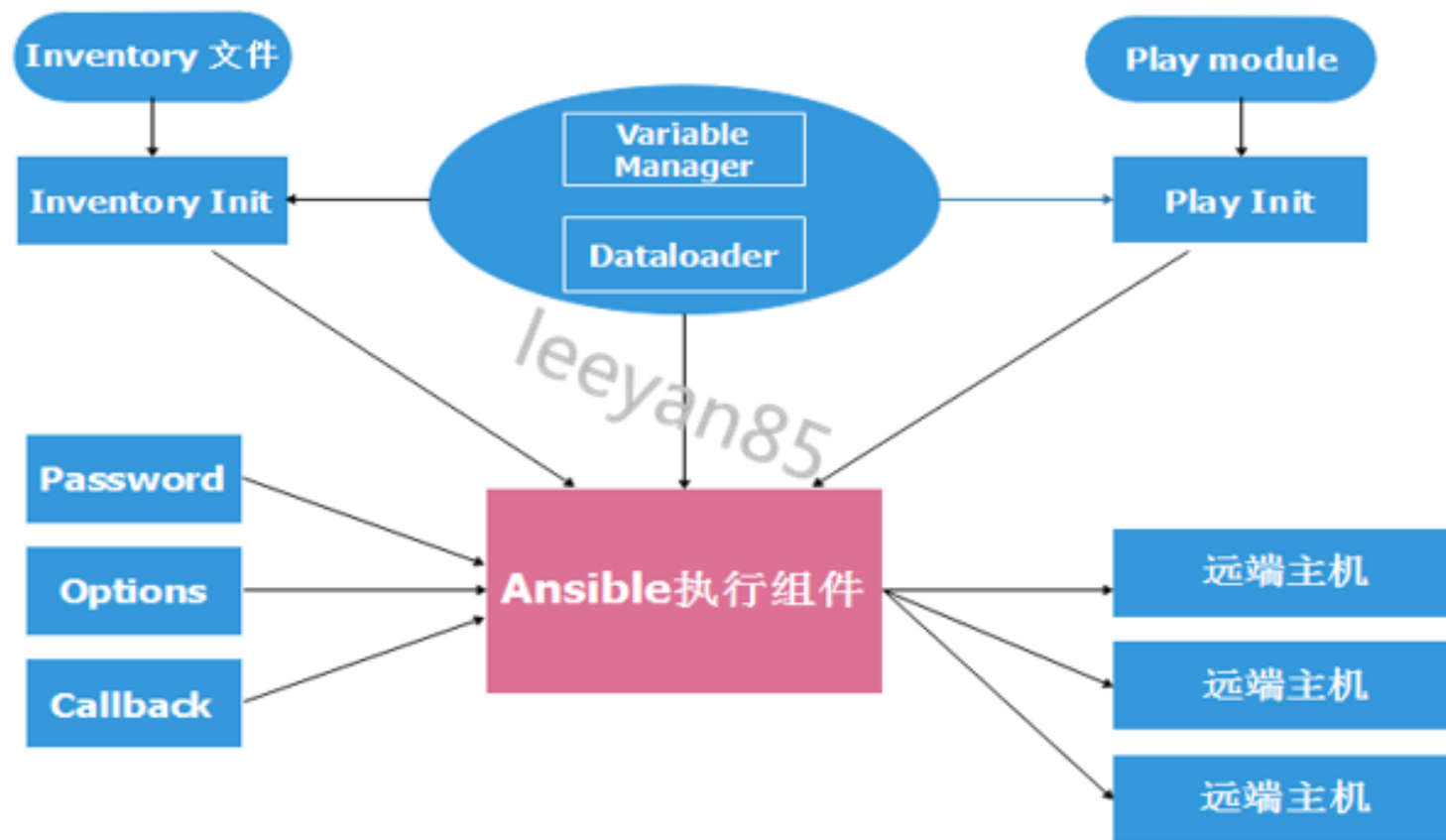
- 配合下图和代码讲解调用流程

- 如何使用Ansible Python API

- 使用Ansible Python API运行ad-hoc task
- 使用Callback Plugin对返回值进行处理
- 使用Ansible Python API运行playbook



# Ansible Python API 的调用流程



# Ansible Plugin

- 什么是Ansible Plugin

首先回顾Ansible Python API的运行过程

- Ansible Plugin有哪些

- Connection
- Lookup
- Vars
- Filter
- Action
- Callback

- 以Callback Plugin为例子讲述如何扩展plugin

