

大型架构及配置技术

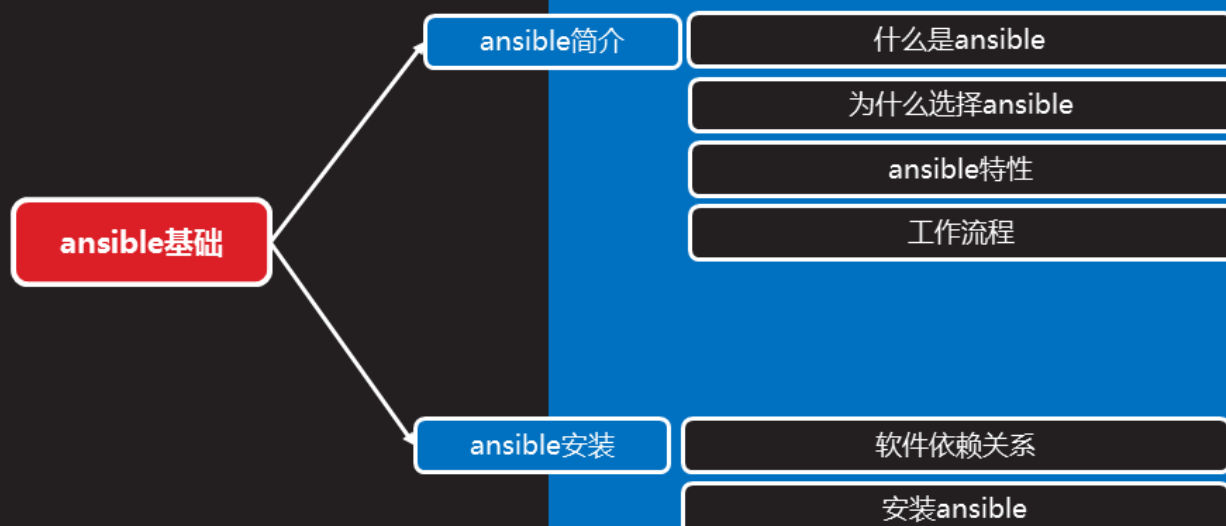
NSD ARCHITECTURE **DAY01**

内容

上午	09:00 ~ 09:30	ansible基础
	09:30 ~ 10:20	
	10:30 ~ 11:20	
	11:30 ~ 12:00	ad-hoc
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	批量配置管理
	16:10 ~ 17:10	
	17:20 ~ 18:00	总结和答疑



ansible基础



ansible简介

什么是ansible

知识讲解

- ansible是2013年推出的一款IT自动化和DevOps软件，2015年被RedHat收购。是基于Python研发，糅合很多老运维工具的优点，实现了批量操作系统配置，批量程序部署，批量运行命令等功能
- ansible可以实现：
 - 自动化部署APP
 - 自动化管理配置项
 - 自动化持续交付
 - 自动化（AWS）云服务管理



为什么选择ansible (续2)



- ansible优点
 - 只需要SSH和Python即可使用
 - 无客户端
 - ansible功能强大，模块丰富
 - 上手容易，门槛低
 - 基于Python开发，做二次开发更容易
 - 使用公司比较多，社区活跃

知识讲解



ansible特性

知识讲解

- 模块化设计，调用特定的模块完成特定任务
- 基于Python语言实现
 - paramiko
 - PyYAML (半结构化语言)
 - Jinja2
- 其模块支持JSON等标准输出格式，可以采用任何编程语言重写



ansible特性（续1）

知识讲解

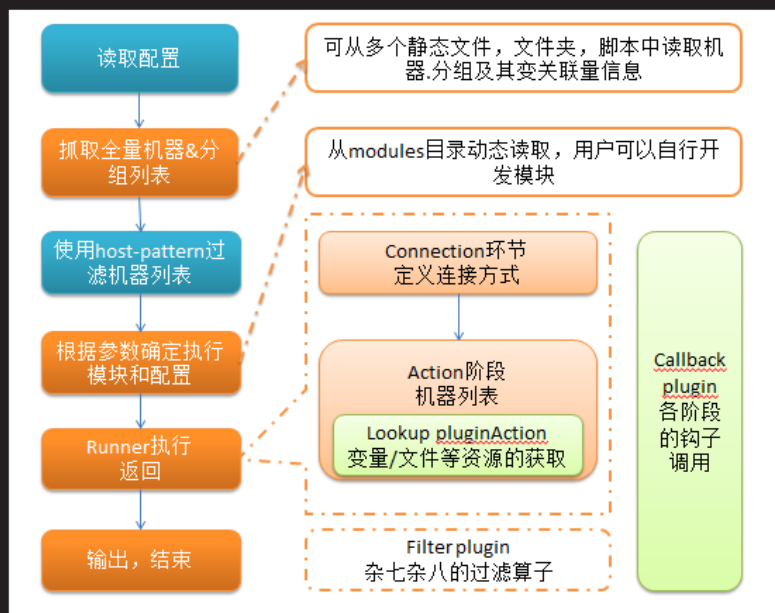
- 部署简单
- 主从模式工作
- 支持自定义模块
- 支持playbook
- 易于使用
- 支持多层部署
- 支持异构IT环境



工作流程

- ansible大体执行过程

知识讲解



ansible安装

软件依赖关系

知识讲解

- 对管理主机
 - 要求Python 2.6 或Python 2.7
- ansible 使用以下模块，都需要安装
 - paramiko
 - PyYAML
 - Jinja2
 - httplib2
 - six



软件依赖关系（续1）

知识讲解

- 对于被托管主机
 - ansible默认通过SSH协议管理机器
 - 被管理主机要开启ssh服务，允许ansible主机登录
 - 在托管节点上也需要安装Python2.5或以上的版本
 - 如果托管节点上开启了SELinux，需要安装libselinux-python



安装ansible

知识讲解

- ansible可以基于源码运行
- 源码安装
 - pip , 需要配置扩展软件包源extras
 - git

```
yum install epel-release  
yum install git python2-pip
```
 - pip安装依赖模块

```
pip install paramiko PyYAML Jinja2 httpplib2 six
```



安装ansible (续1)

知识讲解

- ansible源码下载
 - git clone git://github.com/ansible/ansible.git
 - yum install python-setuptools python-devel
 - python setup.py build
 - python setup.py install
- pip方式安装
 - pip install ansible



安装ansible (续2)

知识讲解

- yum扩展源安装简单，自动解决依赖关系（推荐）
 - <http://mirror.centos.org/.../.../extras/>
 - yum install ansible
- 安装完成以后验证
 - ansible --version

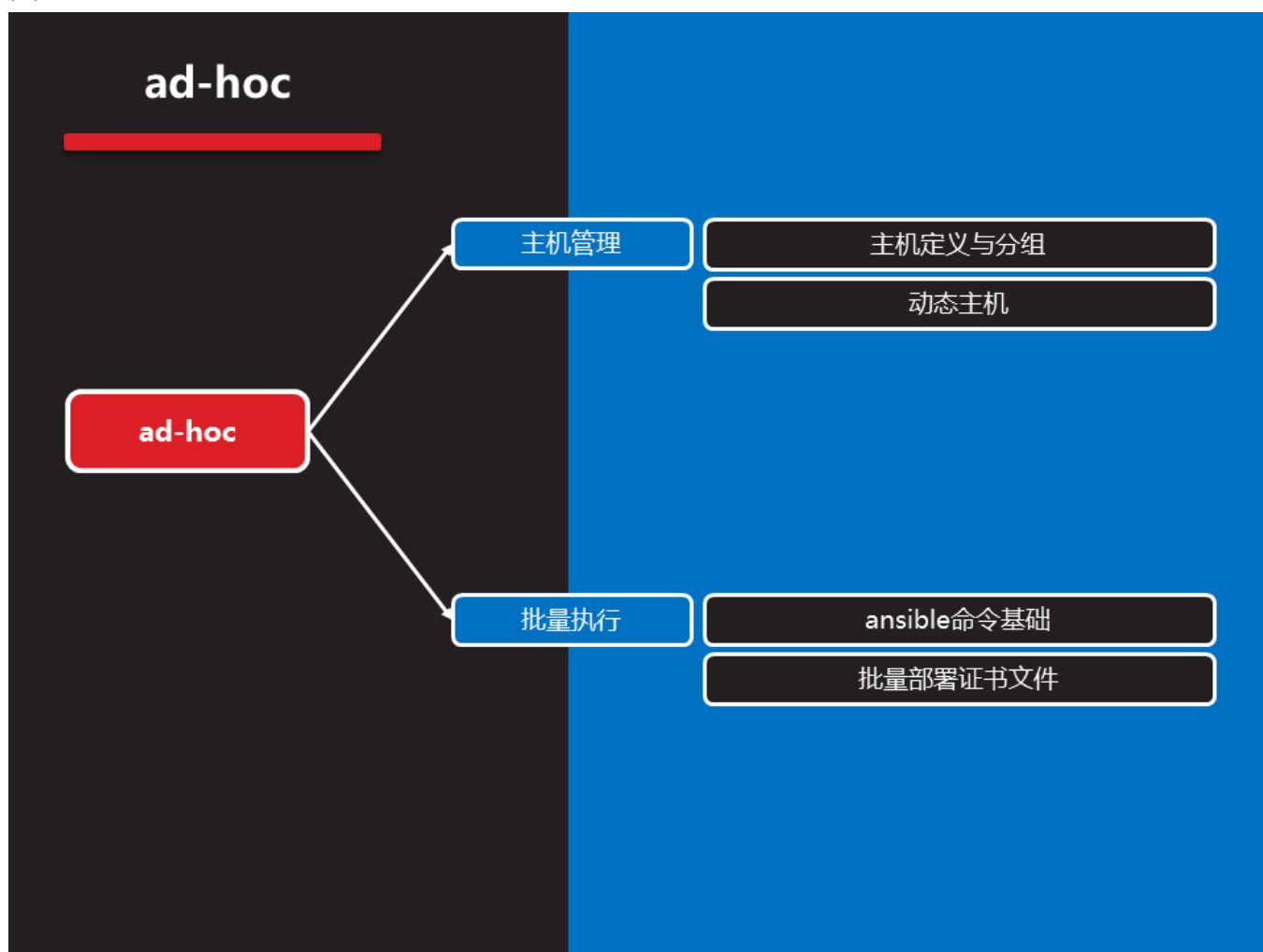


案例1：环境准备

课堂练习

1. 启动6台虚拟机
2. 禁用selinux和firewalld
3. 编辑/etc/hosts
4. 配置yum扩展源并在管理节点安装ansible





主机管理

主机定义与分组（续1）

知识讲解

- ansible.cfg 配置文件
 - inventory定义托管主机地址配置文件
 - 先编辑/etc/ansible/hosts文件，写入远程主机的地址。
- 格式
 - # 表示注释
 - [组名称]
主机名称或ip地址，登录用户名，密码、端口等信息
- 测试
 - ansible [组名称] --list-hosts



主机定义与分组（续2）

知识讲解

- inventory 参数说明
 - ansible_ssh_host
 - 将要连接的远程主机名与你想要设定的主机别名不同，可以通过此变量设置
 - ansible_ssh_port
 - ssh端口号：如果不是默认的端口号，通过此变量设置
 - ansible_ssh_user
 - 默认的ssh用户名



主机定义与分组（续4）

知识讲解

- inventory 参数说明
 - ansible_connection
 - 与主机的连接类型，如：local，ssh或paramiko，1.2以前默认使用paramiko，1.2以后默认使用'smart'，它会根据是否支持ControlPersist来判断'ssh'方式是否可行
 - ansible_ssh_private_key_file
 - ssh使用的私钥文件，适用于有多个密钥，而你不想使用SSH代理的情况



主机定义与分组（续5）

知识讲解

- inventory 参数说明
 - ansible_shell_type
 - 目标系统的shell类型，默认情况下，命令的执行使用'sh'语法，可设置为'csh'或'fish'
 - ansible_python_interpreter
 - 目标主机的python路径，适用情况：系统中有多个Python，或者命令路径不是"/usr/bin/python"



主机定义与分组（续6）

知识讲解

- 分组定义、范围定义样例

```
[web]
```

```
web1
```

```
web2
```

```
[db]
```

```
db[1:2]
```

```
[cache]
```

```
192.168.1.16
```

```
[app1:children]
```

```
web
```

```
db
```



主机定义与分组（续7）

- 分组定义、范围定义样例

```
[web]
```

```
web[1:2]
```

```
[web:vars]
```

```
ansible_ssh_user="root"
```

```
ansible_ssh_pass="pwd "
```

```
ansible_ssh_port="22"
```

```
[cache]
```

```
c01 ansible_ssh_user="root" ansible_ssh_pass="pwd"
```

知识讲解



主机定义与分组（续8）

- 自定义配置文件
 - 创建文件夹myansible
 - 创建配置文件ansible.cfg

```
[defaults]
inventory = myhost
```
 - 配置主机文件

```
[nginx]
192.168.1.11
192.168.1.12
192.168.1.13
```
 - ansible nginx --list-hosts

知识讲解



案例2：主机定义与分组

课堂练习

1. 熟悉ansible配置文件
2. 定义主机，分组和子组练习
3. 自定义文件，多配置路径练习



动态主机

知识讲解

- 无限可能
 - ansible Inventory包含静态和动态的Inventory，静态Inventory指在文件/etc/ansible/hosts中指定的主机和组，动态Inventory指通过外部脚本获取主机列表，按照其要求格式返回给ansilbe命令
- Json
 - JSON (JavaScript Object Notation , JavaScript对象表示法) ，一种基于文本独立于语言的轻量级数据交换格式



动态主机（续1）

知识讲解

- 注意事项：
 - 主机部分必须是列表格式
 - Hostdata行，其中的"hosts" 部分可以省略，但使用时，必须是"hosts"



动态主机（续2）

知识讲解

- 脚本输出主机列表

```
#!/usr/bin/python
import json
hostlist = {}
hostlist["bb"] = ["192.168.1.15", "192.168.1.16"]
hostlist["192.168.1.13"] = {
    "ansible_ssh_user":"root","ansible_ssh_pass":"pwd"
}
hostlist["aa"] = {
    "hosts" : ["192.168.1.11", "192.168.1.12"],
    "vars" : {
        "ansible_ssh_user":"root","ansible_ssh_pass":"pwd"
    }
}
print(json.dumps(hostlist))
```



动态主机（续3）

- 脚本输出样例

```
{
  "aa" : {
    "hosts" : ["192.168.1.11", "192.168.1.12"],
    "vars" : {
      "ansible_ssh_user" : "root",
      "ansible_ssh_pass" : "pwd"
    }
  },
  "bb" : ["192.168.1.15", "192.168.1.16"],
  "192.168.1.13": { "ansible_ssh_user" : "root",
    "ansible_ssh_pass" : "pwd" }
}
```

知识讲解



案例3：动态主机

1. 脚本输出主机列表

课堂练习



批量执行



ansible命令基础（续1）

知识讲解

- 列出要执行的主机
 - `ansible all --list-hosts`
- 批量检测主机
 - `ansible all -m ping`
- 批量执行命令
 - `ansible all -m command -a 'id' -k`

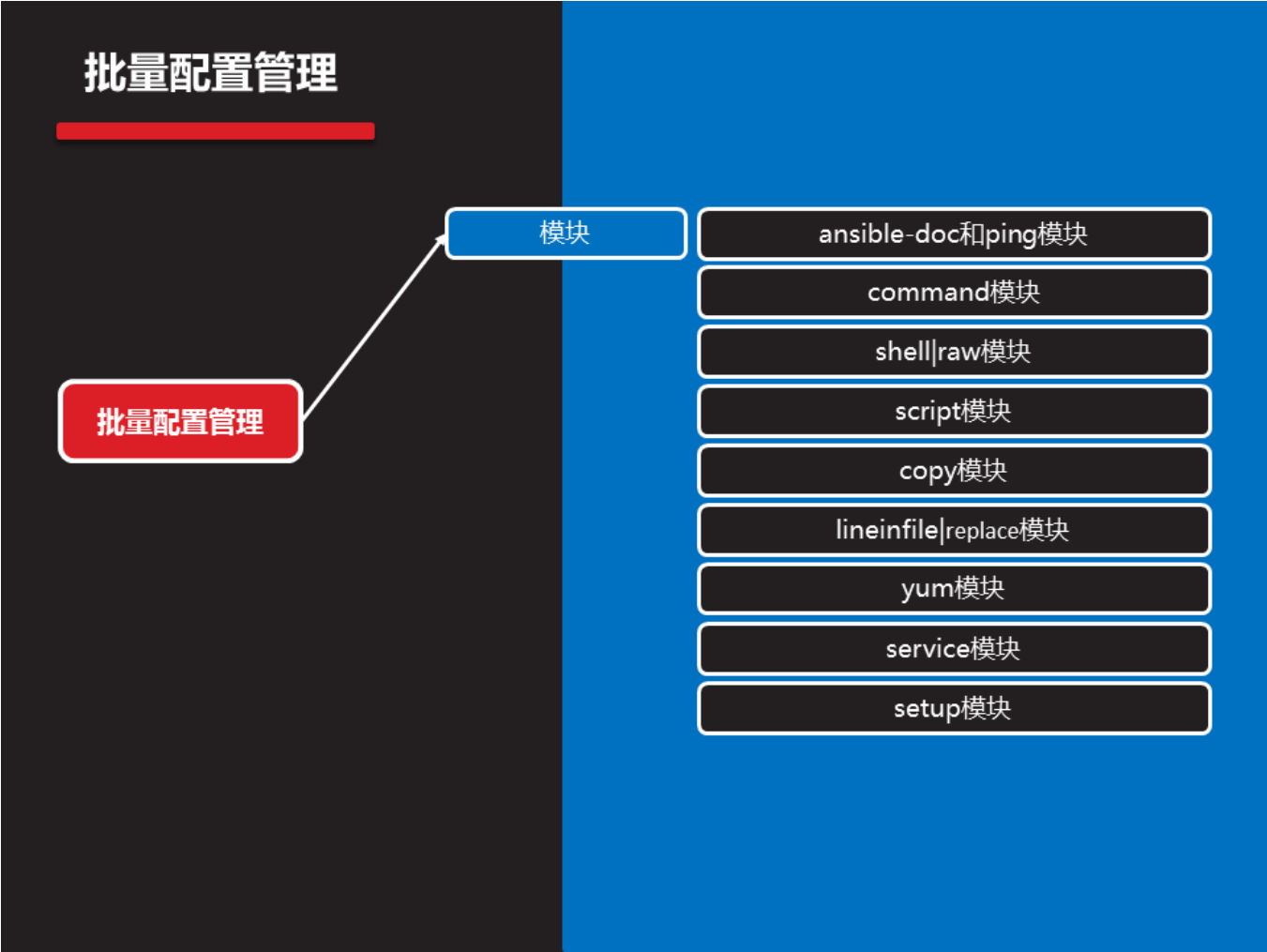


案例4：批量部署证书文件

1. 创建一对密钥
2. 给所有主机部署密钥

课堂练习





Tedu.cn
达内教育

模块

