

## DAY01

### 1 简述什么是 ansible

参考答案

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

### 2 ansible 可以实现什么功能

参考答案

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

### 3 ansible 优点有哪些

参考答案

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

### 4 简述 ansible 中主机定义与分组的步骤

参考答案

1、ansible.cfg 配置文件：

inventory 定义托管主机地址配置文件

2、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

定义主机分组练习

```
[root@ansible ~]# cd /etc/ansible/
[root@ansible ansible]# ls
ansible.cfg  hosts  roles
[root@ansible ansible]# vim ansible.cfg
#inventory      = /etc/ansible/hosts      //指定分组文件路径, 主机的分组文件
hosts
[selinux]        //组名称, selinux 的相关选项在这个下面配置
...
[colors]         //组名称, colors 的相关选项在这个下面配置
...
[root@ansible ansible]# vim hosts
[web]
web1
web2
[root@ansible ansible]# ansible web --list-host    //显示 web 组的主机
hosts (2):
    web1
    web2
```

5 如何用 ansible 批量部署证书文件, 给所有主机部署密钥

参考答案

1) 创建密钥

```
[root@ansible aaa]# cd /root/.ssh/
[root@ansible .ssh]# vi /etc/ansible/hosts
[web]
web1
web2
```

```

[db]
db[1:2]
[other]
cache
[root@ansible .ssh]# ansible all -m ping //直接 ping 会报错
[root@ansible .ssh]# ssh-keygen -t rsa -b 2048 -N "" //创建密钥

```

## 2) 给所有主机部署密钥

```

[root@ansible .ssh]# ansible all -m authorized_key -a "user=root
exclusive=true manage_dir=true key='$(< /root/.ssh/id_rsa.pub)'" -k
SSH password: //输入密码
[root@ansible .ssh]# ansible all -m ping //成功
web2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
db2 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
web1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
cache | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
db1 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
[root@ansible .ssh]# ssh web1 //不需要输入密码,可以直接登陆
Last login: Thu Sep  6 11:49:00 2018 from 192.168.1.51
[root@web1 ~]#

```

## DAY02

### 1 阐述 ansible 七种武器有哪些

参考答案

第一种武器：

**ansible** 命令，用于执行临时性的工作，必须掌握

第二种武器：

**ansible-doc** 是 **ansible** 模块的文档说明，针对每个模块都有详细的说明及

应用案例介绍，功能和 **Linux** 系统 **man** 命令类似，必须掌握

第三种武器：

**ansible-console** 是 **ansible** 为用户提供的交互式工具，用户可以在

**ansible-console** 虚拟出来的终端上像 **Shell** 一样使用 **ansible** 内置的各种命令，这为习惯使用 **Shell** 交互方式的用户提供了良好的使用体验

第四种武器：

**ansible-galaxy** 从 **github** 上下载管理 **Roles** 的一款工具，与 **python** 的

**pip** 类似

第五种武器：

**ansible-playbook** 是日常应用中使用频率最高的命令，工作机制：通过读取

先编写好的 **playbook** 文件实现批量管理，可以理解为按一定条件组成的 **ansible** 任

务集，必须掌握

第六种武器：

**ansible-vault** 主要用于配置文件加密，如编写的 **playbook** 文件中包含敏

感信息，不想其他人随意查看，可用它加密/解密这个文件

第七种武器：

**ansible** 有两种工作模式 **pull/push**，默认使用 **push** 模式工作，**pull** 和

**push** 工作模式机制刚好相反

适用场景：有大批量机器需要配置，即便使用高并发线程依旧要花费很多时间

通常在配置大批量机器的场景下使用，灵活性稍有欠缺，但效率几乎可以无限

提升，对运维人员的技术水平和前瞻性规划有较高要求

## 2 什么是 JSON 以及其特性是什么

参考答案

JSON 简介：

JSON 是 JavaScript 对象表示法，它是一种基于文本独立于语言的轻量级数

据交换格式

JSON 中的分隔符限于单引号 " ' " 、小括号 " ( ) " 、中括号 " [ ] " 、大

括号 " { } " 、冒号 " : " 和逗号 " , "

JSON 特性：

JSON 是纯文本、具有"自我描述性"（人类可读）、具有层级结构（值中存在值）、

可通过 JavaScript 进行解析

## 3 简述 YAML 是什么

参考答案

是一个可读性高，用来表达数据序列的格式

YAML（YAML Ain't Markup Language）参考了多种语言，如：C 语言、Python、Perl 等，并从 XML、电子邮件的数据格式中获得灵感，Clark Evans 在 2001 年首次发表了这种语言，目前已有数种编程语言或脚本语言支持这种语言

## 4 简述什么是 playbook

参考答案

playbook 是 ansible 用于配置，部署和管理托管主机剧本，通过 playbook 的详细描述，执行其中的一系列 tasks，可以让远端主机达到预期状态

也可以说，playbook 字面意思即剧本，现实中由演员按剧本表演，在 ansible 中由计算机进行安装，部署应用，提供对外服务，以及组织计算机处理各种各样的事情

## 5 用 playbook 安装 Apache,修改端口，配置 ServerName，修改主页，设置开机自启

## 参考答案

```
[root@ansible ansible]# vim http.yml
---
- hosts: cache
  remote_user: root
  tasks:
    - name: install one specific version of Apache
      yum:
        name: httpd          //安装 Apache
        state: installed
    - lineinfile:
        path: /etc/httpd/conf/httpd.conf
        regexp: '^Listen '
        line: 'Listen 8080'    //修改端口为 8080
    - replace:
        path: /etc/httpd/conf/httpd.conf
        regexp: '^#(ServerName).*'    //配置 ServerName
        replace: '\1 localhost'
    - service:
        name: httpd
        enabled: yes          //开机自启
        state: restarted
    - copy:
        src: /root/index.html    //修改主页，可以自己写个页面
        dest: /var/www/html/index.html
[root@ansible ansible]# curl 192.168.1.56:8080
hello world
[root@ansible ansible]# ssh cache
Last login: Fri Sep  7 09:32:05 2018 from 192.168.1.51
[root@cache ~]# apachectl -t
Syntax OK
```

## DAY03

### 1 简述什么是 ELK

## 参考答案

ELK 是一整套解决方案，是三个软件产品的首字母缩写，很多公司都在使用，如：Sina、携程、华为、美团等

ELK 分别代表的意思

Elasticsearch: 负责日志检索和储存

Logstash: 负责日志的收集和分析、处理

Kibana: 负责日志的可视化

这三款软件都是开源软件，通常是配合使用，而且又先后归于 Elastic.co 公司名下，故被简称为 ELK

2 ELK 可以实现什么功能

参考答案

在海量日志系统的运维中，可用于解决分布式日志数据集中式查询和管理、系统监控，包含系统硬件和应用各个组件的监控、故障排查、安全信息和事件管理、报表功能

3 Elasticsearch 主要特点

参考答案

1、实时分析

2、分布式实时文件存储，并将每一个字段都编入索引

3、文档导向，所有的对象全部是文档

4、高可用性，易扩展，支持集群（Cluster）、分片和复制（Shards 和 Replicas）

5、接口友好，支持 JSON

4 如何插入，增加，删除和查询数据

参考答案

增加数据

```
[root@se5 ~]# locale
[root@se5 ~]# LANG=en_US.UTF-8 //设置编码
[root@se5 ~]# curl -X PUT "http://192.168.1.65:9200/taindex/teacher/1" -d
'{
  "职业":"诗人",
  "名字":"李白",
  "称号":"诗仙",
  "年代":"唐"
}'
```

```
{"_index":"taindex","_type":"teacher","_id":"1","_version":2,"_shards":{"total":2,"successful":2,"failed":0},"created":false}
```

修改数据

```
[root@se5 ~]# curl -X PUT "http://192.168.1.65:9200/taindex/teacher/1" -d '{
  "doc":{
    "年代": "唐代"
  }
}'
```

```
{"_index":"taindex","_type":"teacher","_id":"1","_version":3,"_shards":{"total":2,"successful":2,"failed":0},"created":false}
```

查询数据

```
[root@se5 ~]# curl -X GET "http://192.168.1.65:9200/taindex/teacher/3?pretty"
{
  "_index" : "taindex",
  "_type" : "teacher",
  "_id" : "3",
  "found" : false
}
```

删除数据

```
[root@se5 ~]# curl -X DELETE "http://192.168.1.65:9200/taindex/teacher/3?pretty"
{
  "found" : false,
  "_index" : "taindex",
  "_type" : "teacher",
  "_id" : "3",
  "_version" : 1,
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  }
}
```



## DAY04

### 1 如何批量导入数据

#### 参考答案

使用\_bulk 批量导入数据

使用 POST 方式批量导入数据，数据格式为 json，url 编码使用 data-binary 导入含有 index 配置的 json 文件

```
[root@room9pc01 ~]# scp /var/ftp/elk/*.gz 192.168.1.66:/root/
[root@kibana ~]# gzip -d logs.jsonl.gz
[root@kibana ~]# gzip -d accounts.json.gz
[root@kibana ~]# gzip -d shakespeare.json.gz
[root@kibana ~]# curl -X POST "http://192.168.1.61:9200/_bulk" \
--data-binary @shakespeare.json
[root@kibana ~]# curl -X POST "http://192.168.1.61:9200/xixi/haha/_bulk" \
--data-binary @accounts.json
//索引是 xixi，类型是 haha，必须导入索引和类型，没有索引，要加上
[root@kibana ~]# curl -X POST "http://192.168.1.61:9200/_bulk" \
--data-binary @logs.jsonl
```

### 2 如何实现数据批量查询

#### 参考答案

```
[root@kibana ~]# curl -XGET 'http://192.168.1.61:9200/_mget?pretty' -d '{
  "docs":[
    {
      "_index":"shakespeare",
      "_type":"act",
      "_id":0
    },
    {
      "_index":"shakespeare",
      "_type":"line",
      "_id":0
    },
    {
      "_index":"xixi",
```

```

        "_type": "haha",
        "_id": 25
    }
]
}'
{
    //查询的结果
    "docs": [ {
        "_index": "shakespeare",
        "_type": "act",
        "_id": "0",
        "_version": 1,
        "found": true,
        ...
        ...
        "employer": "Filodyne",
        "email": "virginiaayala@filodyne.com",
        "city": "Nicholson",
        "state": "PA"
    }
    ]
}

```

### 3 map 映射的作用及分类是什么

参考答案

作用：这样会让索引建立得更加的细致和完善

分类：静态映射和动态映射

动态映射：自动根据数据进行相应的映射

静态映射：自定义字段映射数据类型

### 4 Logstash 插件有哪些

参考答案

codec 类插件、file 插件、tcp 和 udp 插件、syslog 插件、filter grok 插件

## DAY05

### 1 大数据的由来

参考答案

随着计算机技术的发展，互联网的普及，信息的积累已经到了一个非常庞大的地步，信息的增长也在不断的加快，随着互联网、物联网建设的加快，信息更是爆炸式增长，收集、检索、统计这些信息越发困难，必须使用新的技术来解决这些问题

## 2 什么是大数据

参考答案

数据指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合，需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产

是指从各种各样类型的数据中，快速获得有价值的信息

## 3 简述大数据特性有哪些

参考答案

**Volume (大体量):** 可从数百 TB 到数十数百 PB、甚至 EB 的规模

**Variety (多样性):** 大数据包括各种格式和形态的数据

**Velocity (时效性):** 很多大数据需要在一定的时间限度下得到及时处理

**Veracity (准确性):** 处理的结果要保证一定的准确性

**Value (大价值):** 大数据包含很多深度的价值，大数据分析挖掘和利用将带来巨大的商业价值

## 4 Hadoop 常用组件以及核心组件有哪些

参考答案

**HDFS:** Hadoop 分布式文件系统（核心组件）

**MapReduce:** 分布式计算框架（核心组件）

**Yarn:** 集群资源管理系统（核心组件）

**Zookeeper:** 分布式协作服务

**Hbase:** 分布式列存数据库

**Hive:** 基于 Hadoop 的数据仓库

**Sqoop:** 数据同步工具

**Pig:** 基于 Hadoop 的数据流系统

**Mahout:** 数据挖掘算法库

**Flume:** 日志收集工具

**5 Hadoop 如何实现统计词频**

参考答案

```
[root@nn01 ~]# cd /usr/local/hadoop/
[root@nn01 hadoop]# mkdir /usr/local/hadoop/aa
[root@nn01 hadoop]# ls
bin    etc    include  lib    libexec  LICENSE.txt  NOTICE.txt  aa
README.txt  sbin  share
[root@nn01 hadoop]# cp *.txt /usr/local/hadoop/aa
[root@nn01 hadoop]# ./bin/hadoop jar \
share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.6.jar
wordcount aa bb          //wordcount 为参数 统计 aa 这个文件夹，存到 bb 这个文件
里面（这个文件不能存在，要是存在会报错，是为了防止数据覆盖）
[root@nn01 hadoop]# cat bb/part-r-00000    //查看
```

**DAY06**

**1 如何查看 Hadoop 集群文件系统的根以及怎么创建**

参考答案

查看

```
[root@nn01 hadoop]# /usr/local/hadoop/bin/hadoop fs -ls /
```

创建

```
[root@nn01 hadoop]# /usr/local/hadoop/bin/hadoop fs -mkdir /aaa
```

**2 如何增加一个新的节点**

参考答案

**1) 增加一个新的节点 node4**

```

[root@hadoop5 ~]# echo node4 > /etc/hostname //更改主机名为 node4
[root@hadoop5 ~]# hostname node4
[root@node4 ~]# yum -y install rsync
[root@node4 ~]# yum -y install java-1.8.0-openjdk-devel
[root@node4 ~]# mkdir /var/hadoop
[root@nn01 .ssh]# ssh-copy-id 192.168.1.25
[root@nn01 .ssh]# vim /etc/hosts
192.168.1.21 nn01
192.168.1.22 node1
192.168.1.23 node2
192.168.1.24 node3
192.168.1.25 node4
[root@nn01 .ssh]# scp /etc/hosts 192.168.1.25:/etc/
[root@nn01 ~]# cd /usr/local/hadoop/
[root@nn01 hadoop]# vim ./etc/hadoop/slaves
node1
node2
node3
node4
[root@nn01 hadoop]# for i in {22..25}; do rsync -aSH --delete
/usr/local/hadoop/
\ 192.168.1.$i:/usr/local/hadoop/ -e 'ssh' & done //同步配置
[1] 1841
[2] 1842
[3] 1843
[4] 1844
[root@node4 hadoop]# ./sbin/hadoop-daemon.sh start datanode //启动
2) 查看状态
[root@node4 hadoop]# jps
24439 Jps
24351 DataNode

```

### 3 如何设置同步带宽

参考答案

```

[root@node4 hadoop]# ./bin/hdfs dfsadmin -setBalancerBandwidth
60000000
Balancer bandwidth is set to 60000000
[root@node4 hadoop]# ./sbin/start-balancer.sh

```

### 4 启动 NFS 需要注意什么

## 参考答案

portmap 服务只能用 root 用户启动，nfs3 只能用代理用户启动，用 root 用户执行 jps 可以看到 portmap 和 nfs3，代理用户执行 jps 看不到 portmap

```
[root@nfsgw ~]# /usr/local/hadoop/sbin/hadoop-daemon.sh
--script ./bin/hdfs start portmap //portmap 服务只能用 root 用户启动
starting portmap, logging to
/usr/local/hadoop/logs/hadoop-root-portmap-nfsgw.out
[root@nfsgw ~]# jps
23714 Jps
23670 Portmap
[root@nfsgw ~]# su - nfs
Last login: Mon Sep 10 12:31:58 CST 2018 on pts/0
[nfs@nfsgw ~]$ cd /usr/local/hadoop/
[nfs@nfsgw hadoop]$ ./sbin/hadoop-daemon.sh --script ./bin/hdfs start
nfs3
//nfs3 只能用代理用户启动
starting nfs3, logging to /usr/local/hadoop/logs/hadoop-nfs-nfs3-nfsgw.out
[nfs@nfsgw hadoop]$ jps
1362 Jps
1309 Nfs3
[root@nfsgw hadoop]# jps //root 用户执行可以看到 portmap 和
nfs3
1216 Portmap
1309 Nfs3
1374 Jps
```

## DAY07

### 1 简述什么是 Zookeeper 以及其作用

#### 参考答案

什么是 Zookeeper:

Zookeeper 是一个开源的分布式应用程序协调服务

作用:

用来保证数据在集群间的事务一致性

### 2 简述 Zookeeper 角色与特性

参考答案

**Leader:** 接受所有 **Follower** 的提案请求并统一协调发起提案的投票，负责与所有的 **Follower** 进行内部数据交换

**Follower:** 直接为客户端服务并参与提案的投票，同时与 **Leader** 进行数据交换

**Observer:** 直接为客户端服务但并不参与提案的投票，同时也与 **Leader** 进行数据交换

3 如何利用 **api** 查看 **Zookeeper** 的状态

参考答案

```
[root@nn01 conf]# /usr/local/zookeeper/bin/zkServer.sh start
[root@nn01 conf]# vim api.sh
#!/bin/bash
function getstatus(){
    exec 9<>/dev/tcp/$1/2181 2>/dev/null
    echo stat >&9
    MODE=$(cat <&9 |grep -Po "(?<=Mode:).*")
    exec 9<&-
    echo ${MODE:-NULL}
}
for i in node{1..3} nn01;do
    echo -ne "${i}\t"
    getstatus ${i}
done
[root@nn01 conf]# chmod 755 api.sh
[root@nn01 conf]# ./api.sh
node1    follower
node2    leader
node3    follower
nn01     observer
```

4 如何在 **Kafka** 集群里创建一个 **topic**

参考答案

```
[root@node1 local]# /usr/local/kafka/bin/kafka-topics.sh --create
--partitions 1 --replication-factor 1 --zookeeper node3:2181 --topic aa
Created topic "aa".
```