

大型架构及配置技术

NSD ARCHITECTURE

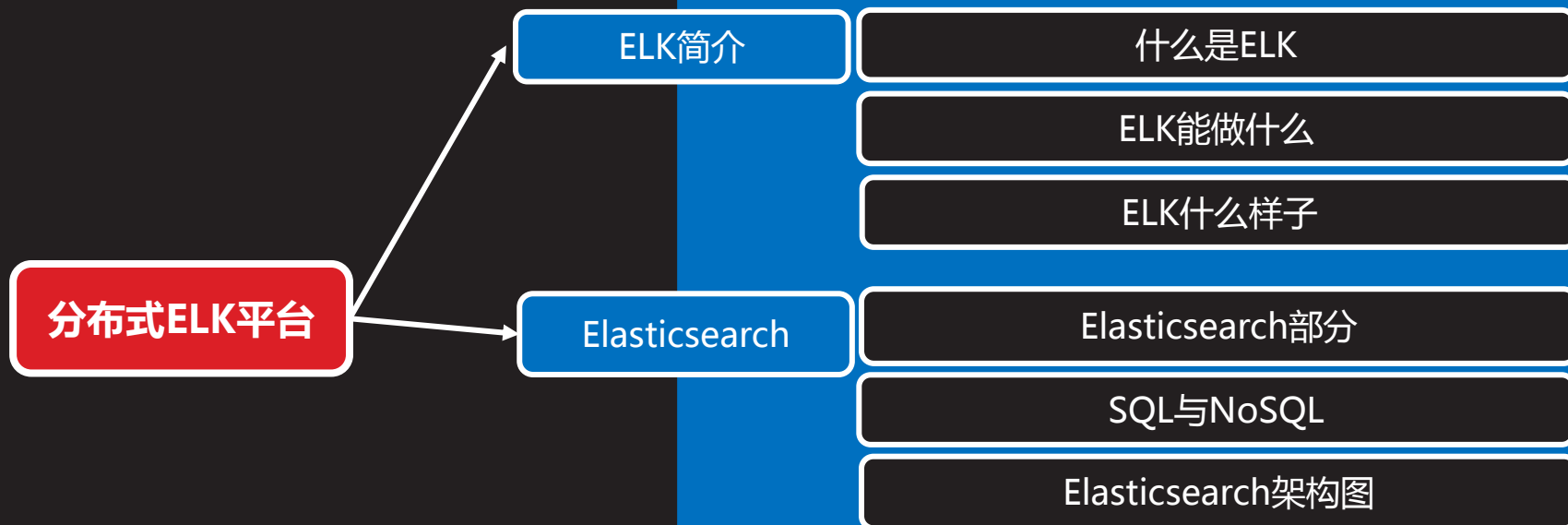
DAY03

内容

| | | |
|----|---------------|----------|
| 上午 | 09:00 ~ 09:30 | 作业讲解和回顾 |
| | 09:30 ~ 10:20 | 分布式ELK平台 |
| | 10:30 ~ 11:20 | ES集群安装 |
| | 11:30 ~ 12:00 | |
| 下午 | 14:00 ~ 14:50 | 扩展插件 |
| | 15:00 ~ 15:50 | |
| | 16:10 ~ 17:10 | Kibana安装 |
| | 17:20 ~ 18:00 | 总结和答疑 |



分布式ELK平台



ELK简介



什么是ELK

- 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
 - 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

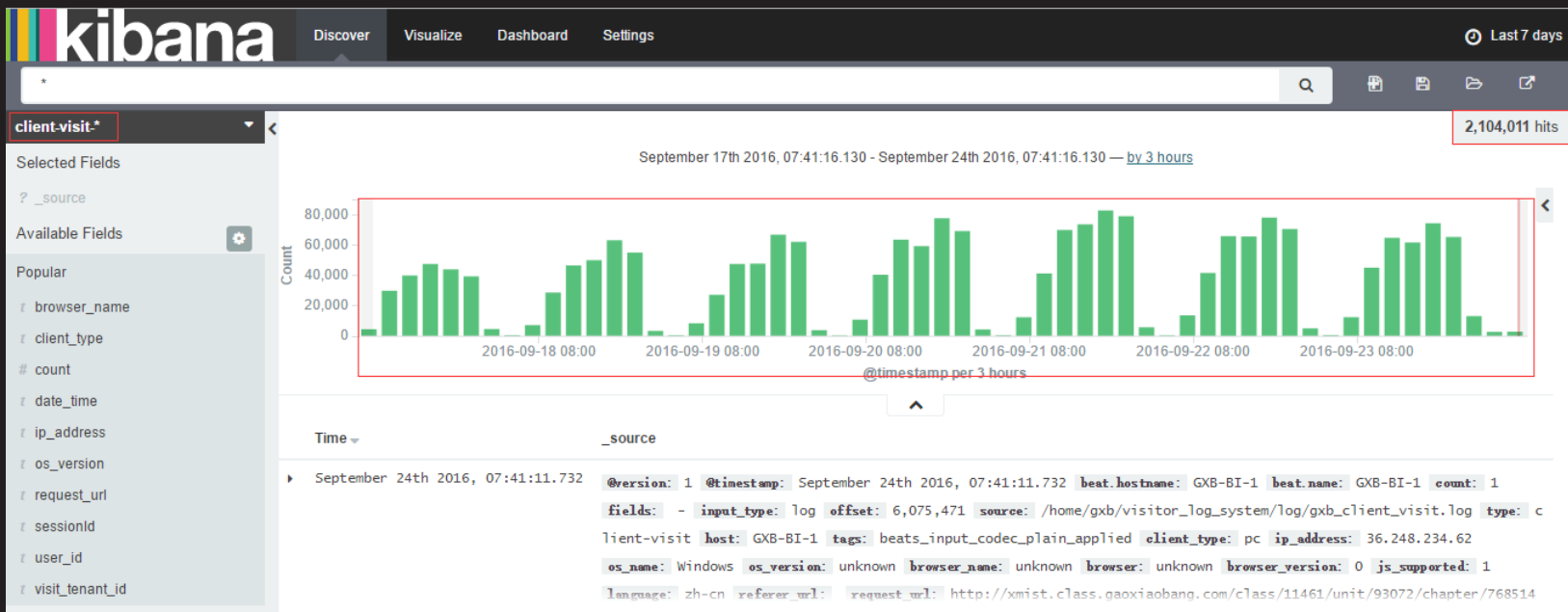


ELK能做什么

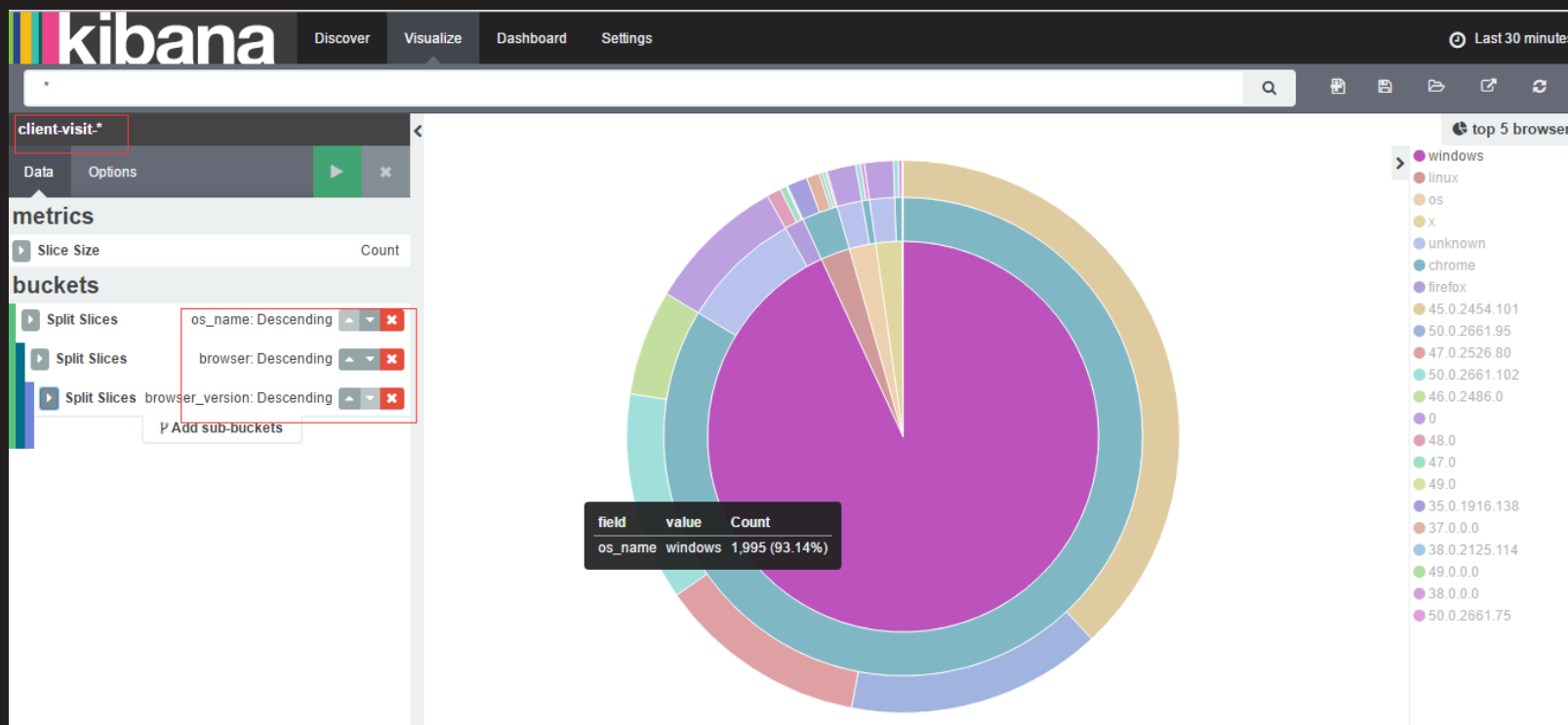
- ELK组件在海量日志系统的运维中，可用于解决
 - 分布式日志数据集中式查询和管理
 - 系统监控，包含系统硬件和应用各个组件的监控
 - 故障排查
 - 安全信息和事件管理
 - 报表功能



ELK什么样子

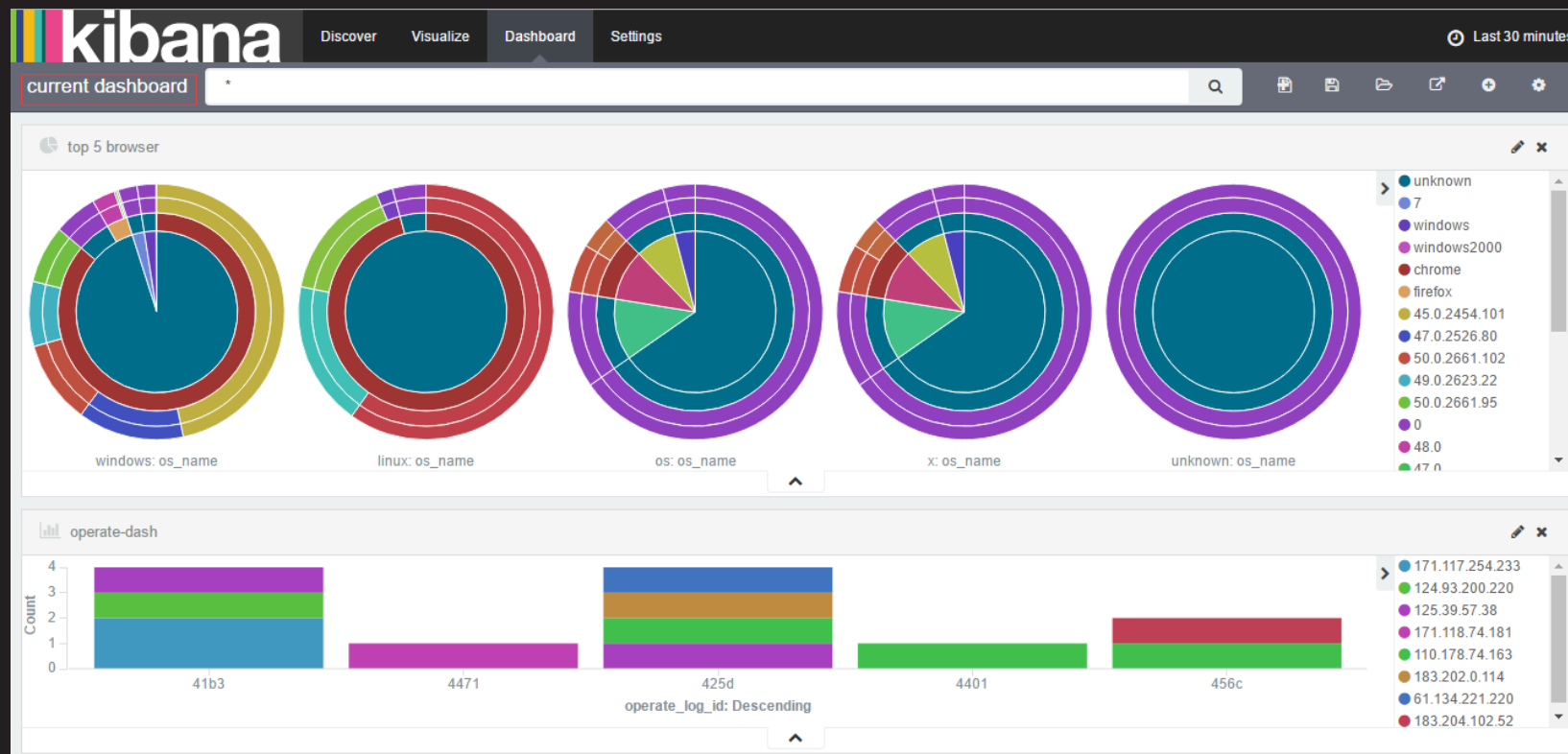


ELK什么样子（续1）



ELK什么样子（续2）

知识讲解



Elasticsearch

Elasticsearch部分

- ElasticSearch是一个基于Lucene的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于RESTful API的Web接口
- Elasticsearch是用Java开发的，并作为Apache许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够达到实时搜索，稳定，可靠，快速，安装使用方便



Elasticsearch部分（续1）

- 主要特点
 - 实时分析
 - 分布式实时文件存储，并将每一个字段都编入索引
 - 文档导向，所有的对象全部是文档
 - 高可用性，易扩展，支持集群（Cluster）、分片和复制（Shards 和 Replicas）
 - 接口友好，支持JSON



Elasticsearch部分（续2）

- ES没有什么
 - Elasticsearch没有典型意义的事务
 - Elasticsearch是一种面向文档的数据库
 - Elasticsearch没有提供授权和认证特性



Elasticsearch部分（续3）

- 相关概念
 - Node：装有一个ES服务器的节点
 - Cluster：有多个Node组成的集群
 - Document：一个可被搜索的基础信息单元
 - Index：拥有相似特征的文档的集合
 - Type：一个索引中可以定义一种或多种类型
 - Field：是ES的最小单位，相当于数据的某一行
 - Shards：索引的分片，每一个分片就是一个Shard
 - Replicas：索引的拷贝



SQL与NoSQL

• ES与关系型数据库的对比

- 在ES中，文档归属于一种 类型（ type ），而这些类型存在于索引（ index ）中，类比传统关系型数据库
- DB -> Databases -> Tables -> Rows -> Columns
- 关系型 数据库 表 行 列
- ES -> Indices -> Types -> Documents -> Fields
- ES 索引 类型 文档 域（ 字段 ）



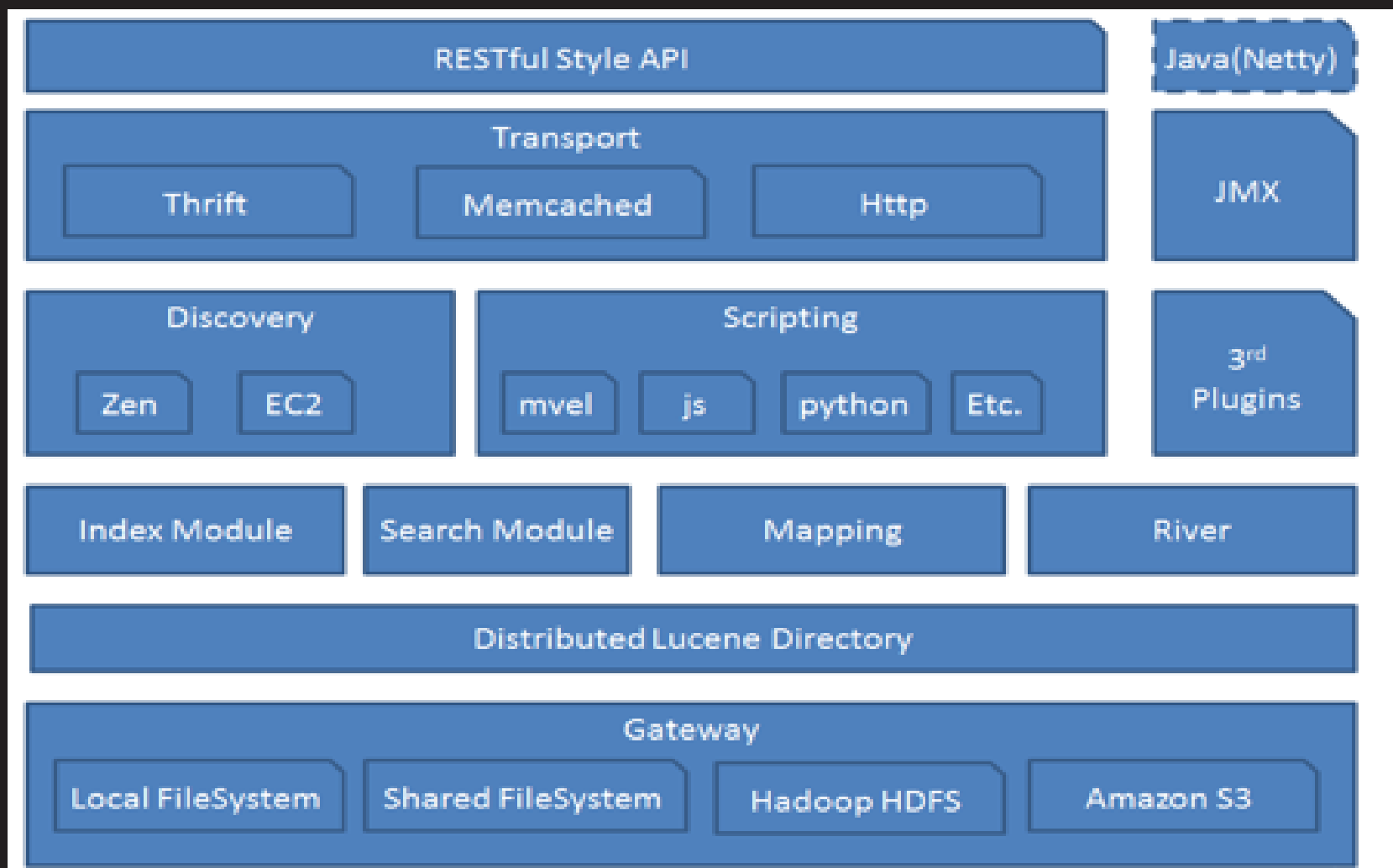
SQL与NoSQL (续1)

- ES与关系型数据库的对比

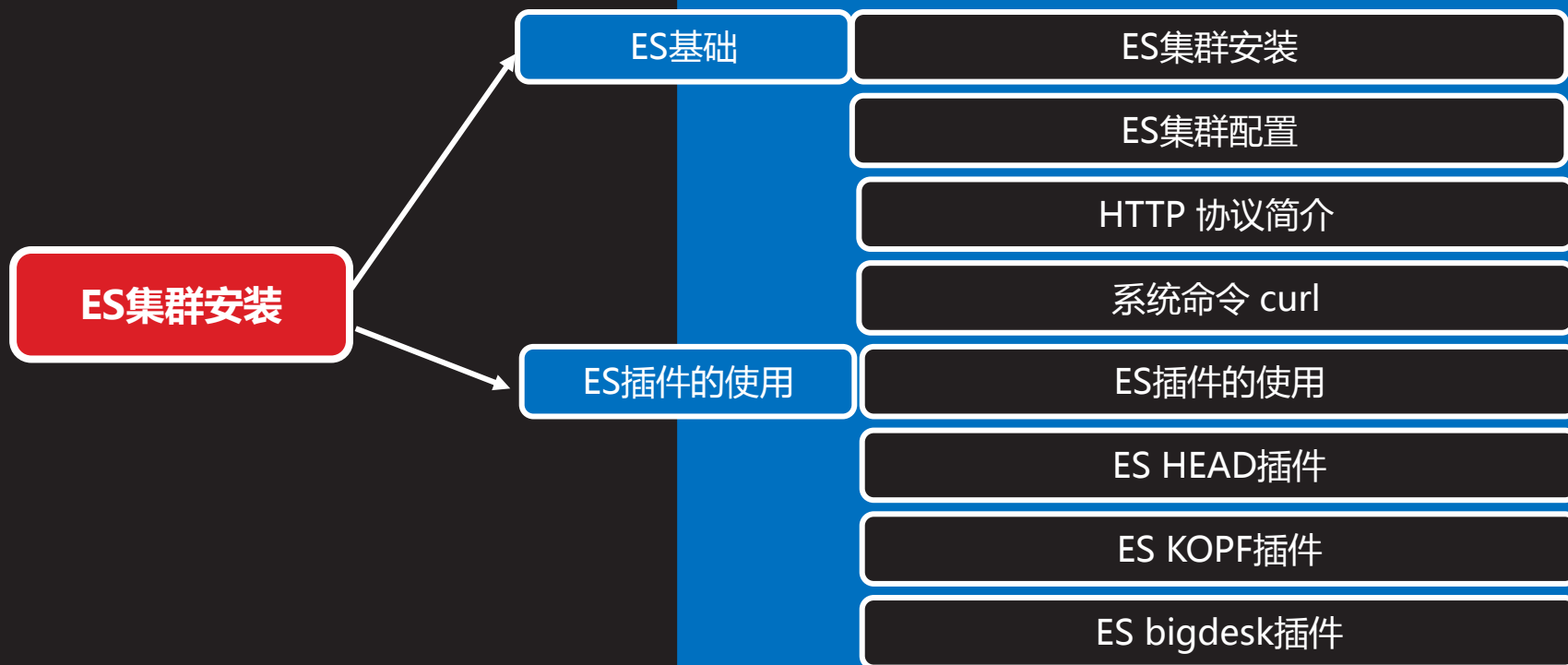
| <i>Relational database</i> | <i>Elasticsearch</i> |
|----------------------------|-----------------------|
| Database | Index |
| Table | Type |
| Row | Document |
| Column | Field |
| Schema | Mapping |
| Index | Everything is indexed |
| SQL | Query DSL |
| SELECT * FROM table... | GET http://... |
| UPDATE table SET | PUT http://... |



Elasticsearch架构图



ES集群安装



ES基础

ES集群安装

- 安装第一台ES服务器
 - 设置主机名称和ip对应关系
 - 解决依赖关系
 - 安装软件包
 - 修改配置文件
 - 启动服务
 - 检查服务



ES集群安装（续1）

- 设置ip与主机名称对应关系
 - 配置/etc/hosts
192.168.4.11 node1
- 安装JDK
 - Elasticsearch要求至少Java 7
 - 一般推荐使用OpenJDK 1.8
 - 配置好安装源以后，我们先解决依赖关系
yum install -y java-1.8.0-openjdk



ES集群安装（续2）

- 安装ES

```
rpm -ivh elasticsearch-2.3.4-1.noarch
```

- 修改配置文件

- elasticsearch.yml

```
network.host: 0.0.0.0
```



ES集群安装（续3）

- 启动服务

- 启动服务并设开机自启

- `systemctl enable elasticsearch`
`systemctl start elasticsearch`

- 验证：

- `netstat -ltunp`

- 能够看到9200，9300被监听



ES集群安装（续4）

- 通过浏览器或curl访问9200端口

```
curl http://192.168.4.11:9200/
{
  "name" : "node1",
  "cluster_name" : "my-es",
  "version" : {
    "number" : "2.3.4",
    .....
    "build_snapshot" : false,
    "lucene_version" : "5.5.0"
  },
  "tagline" : "You Know, for Search"
}
```



案例1：ES集群安装

1. 准备1台虚拟机
2. 部署elasticsearch第一个节点
3. 访问9200端口查看是否安装成功



ES集群配置

- ES集群配置
 - ES集群配置也很简单，只需要对配置文件做少量的修改即可，其他步骤和单机完全一致
 - ES集群配置文件

```
cluster.name: my-es  
node.name: node1  
network.host: 0.0.0.0  
discovery.zen.ping.unicast.hosts: ["node1", "node2",  
"node3"]
```



ES集群配置（续1）

- ES集群配置
 - 集群中的所有节点要相互能够ping通，要在所有集群机器上配置/etc/hosts中的主机名与ip对应关系
 - 集群中所有机器都要安装Java环境
 - cluster.name集群名称配置要求完全一致
 - node.name为当前节点标识，应配置本机的主机名
 - discovery为集群节点机器，不需要全部配置
 - 配置完成以后启动所有节点服务



ES集群配置（续2）

- ES集群配置

- 验证集群，使用ES内置字段 `_cluster/health`

```
curl http://192.168.4.11:9200/_cluster/health?pretty
{
  "cluster_name" : "my-es",
  "status" : "green",
  .....
  "number_of_nodes" : 5,
  "number_of_data_nodes" : 5,
  .....
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 100.0
}
```



ES集群配置（续3）

- ES 集群验证
 - 返回字段解析
 - status " : " green " 集群状态，绿色为正常，黄色表示有问题但不是很严重，红色表示严重故障
 - "number_of_nodes" : 5, 表示集群中节点的数量
 - "number_of_data_nodes" : 5,
 -
 - "task_max_waiting_in_queue_millis" : 0,
 - "active_shards_percent_as_number" : 100.0



案例2：ES集群安装配置

1. 一共安装5台虚拟机
2. 在所有机器中部署ES
3. 启动服务查看验证集群状态



HTTP 协议简介

- http请求由三部分组成
 - 分别是：请求行、消息报头、请求正文
 - 请求行以一个方法符号开头，以空格分开，后面跟着请求的URI和协议版本，格式如下：
Method Request-URI HTTP-Version CRLF



HTTP 协议简介（续1）

- http请求方法
 - 常用方法 GET , POST , HEAD
 - 其他方法 OPTIONS , PUT , DELETE , TRACE和 CONNECT
- ES 常用
 - PUT --- 增
 - DELETE --- 删
 - POST --- 改
 - GET --- 查



系统命令 curl

- 在linux中curl是一个利用URL规则在命令行下工作的文件传输工具，可以说是一款很强大的http命令行工具。它支持多种请求模式，自定义请求头等强大功能，是一款综合工具
- curl 常用参数介绍
 - -A 修改请求 agent
 - -X 设置请求方法
 - -i 显示返回头信息



案例3：练习curl命令

1. 练习使用curl命令
2. 理解GET POST
3. 使用curl命令访问ES集群



ES插件的使用



ES插件的使用

- ES常用插件
- head插件
 - 它展现ES集群的拓扑结构，并且可以通过它来进行索引（Index）和节点（Node）级别的操作
 - 它提供一组针对集群的查询API，并将结果以json和表格形式返回
 - 它提供一些快捷菜单，用以展现集群的各种状态



ES插件的使用（续1）

- ES常用插件
- Kopf插件
 - 是一个ElasticSearch的管理工具
 - 它提供了对ES集群操作的API
- Bigdesk插件
 - 是elasticsearch的一个集群监控工具
 - 可以通过它来查看es集群的各种状态，如：cpu、内存使用情况，索引数据、搜索情况，http连接数等



ES插件的使用（续2）

- ES插件安装、查看

- 查看安装的插件

```
/usr/share/elasticsearch/bin/plugin list
```

- 安装插件

```
/usr/share/elasticsearch/bin/plugin install
```

```
ftp://192.168.4.254/head.zip
```

```
/usr/share/elasticsearch/bin/plugin install
```

```
file:///tmp/kopf.zip
```

- 这里必须使用 url 的方式进行安装，如果文件在本地，我们也需要使用 file:// 的方式指定路径，例如文件在 /tmp/xxx 下面，我们要写成 file:///tmp/xxx，删除使用 remove 指令



ES HEAD插件



ES KOPF插件

cluster

nodes

rest

more

my-es @ node5

5 nodes

2 indices

12 shards

14,006 docs

91.23MB

filter nodes by name

☒ master

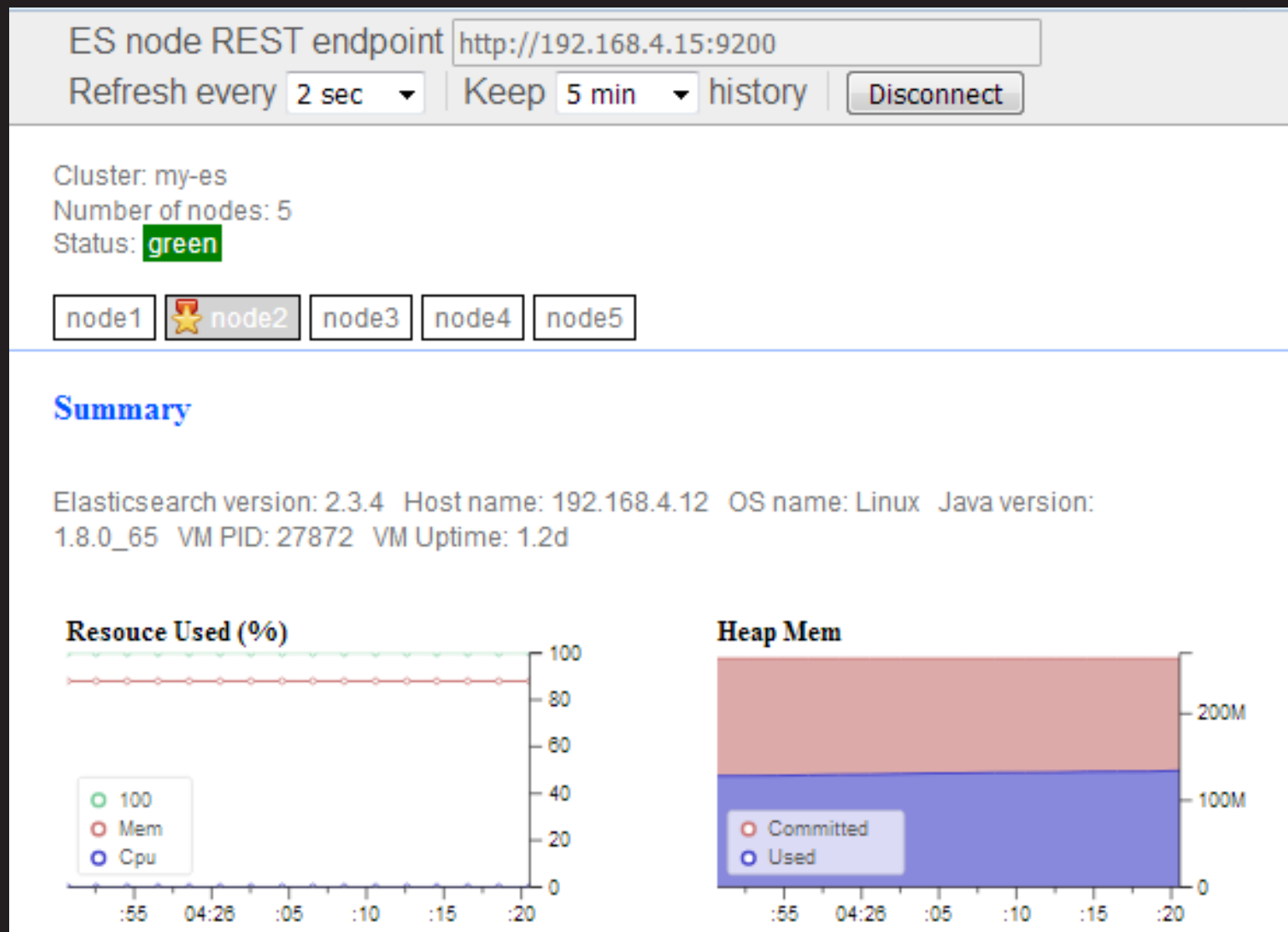
☒ data

☒ client

| name ^ | load average | cpu % | heap usage % | disk usage % | uptime |
|--|--------------|-------|--|--|--------|
| <div>☆ node1</div> <div>192.168.4.11</div> <div>192.168.4.11:9300</div> <div>JVM: 1.8.0_65 ES: 2.3.4</div> | N/A | 0.0 | 16.0 used: 170.30MB max: 1007.38MB | 9.0 free: 15.94GB total: 17.46GB | 2h. |
| <div>★ node2</div> <div>192.168.4.12</div> <div>192.168.4.12:9300</div> <div>JVM: 1.8.0_65 ES: 2.3.4</div> | 0.0 | 0.0 | 10.0 used: 109.82MB max: 1007.38MB | 8.0 free: 16.06GB total: 17.46GB | 1d. |
| <div>☆ node3</div> <div>192.168.4.13</div> <div>192.168.4.13:9300</div> <div>JVM: 1.8.0_65 ES: 2.3.4</div> | N/A | 0.0 | 20.0 used: 206.30MB max: 1007.38MB | 8.0 free: 16.06GB total: 17.46GB | 1d. |
| <div>☆ node4</div> <div>192.168.4.14</div> <div>192.168.4.14:9300</div> <div>JVM: 1.8.0_65 ES: 2.3.4</div> | N/A | 0.0 | 21.0 used: 217.80MB max: 1007.38MB | 8.0 free: 16.06GB total: 17.46GB | 1d. |
| <div>☆ node5</div> <div>192.168.4.15</div> <div>192.168.4.15:9300</div> <div>JVM: 1.8.0_65 ES: 2.3.4</div> | N/A | 0.0 | 11.0 used: 116.56MB max: 1007.38MB | 8.0 free: 16.04GB total: 17.46GB | 1d. |



ES bigdesk 插件



案例4：练习插件

1. 在其中一台机器上部署插件
2. 使用bigdesk查看集群状态
3. 使用head创建index
4. 使用kopf查看数据



扩展插件



RESTful API



RESTful API 调用

- Elasticsearch提供了一系列RESTful的API
 - 检查集群、节点、索引的健康度、状态和统计
 - 管理集群、节点、索引的数据及元数据
 - 对索引进行CRUD操作及查询操作
 - 执行其他高级操作如分页、排序、过滤等
- POST或PUT数据使用json格式



RESTful API 调用（续1）

- JSON
 - JSON (JavaScript Object Notation) , 意思是 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 传输的就是一个字符串
 - Python 中对应的字符串, 列表, 字典都可以转换成对应的 JSON 格式



RESTful API 调用 (续2)

- RESTful API的简单使用
 - _cat API查询集群状态，节点信息
 - v参数显示详细信息
http://192.168.4.15:9200/_cat/health?v
 - help显示帮助信息
http://192.168.4.15:9200/_cat/health?help



RESTful API 调用 (续3)

- Rest API 的简单使用
 - nodes 查询节点状态信息
http://192.168.4.15:9200/_cat/nodes?v
 - 索引信息
http://192.168.4.15:9200/_cat/indices?v



RESTful API 调用 (续4)

- RESTful API增加
 - 创建一个索引，并设置分片数量与副本数量

```
curl -XPUT 'http://192.168.4.13:9200/tarena/' -d '{
  "settings":{
    "index":{
      "number_of_shards": 5,
      "number_of_replicas": 1
    }
  }
}'
```



RESTful API 调用 (续5)

- RESTful API插入数据

```
curl -XPUT 'http://192.168.4.11:9200/tarena/teacher/1' -d '{
  "职业": "诗人",
  "名字": "李白",
  "称号": "诗仙",
  "年代": "唐"
}'
```

```
curl -XPUT 'http://192.168.4.11:9200/tarena/teacher/2' -d '{
  "职业": "诗人",
  "名字": "杜甫",
  "称号": "诗圣",
  "年代": "唐"
}'
```



RESTful API 调用 (续6)

- RESTful API插入数据

```
curl -XPUT 'http://192.168.4.11:9200/tarena/teacher/3' -d '{
  "职业": "诗人",
  "名字": "白居易",
  "称号": "诗魔",
  "年代": "唐"
}'
```

```
curl -XPUT 'http://192.168.4.11:9200/tarena/teacher/4' -d '{
  "职业": "诗人",
  "名字": "李贺",
  "称号": "诗鬼",
  "年代": "唐"
}'
```



RESTful API 调用 (续7)

- POST修改

```
curl -XPOST
'http://192.168.4.11:9200/tarena/teacher/3/_update' -d '{
    "doc":{
        "年代": "唐代"
    }
}'
```

- 查询与删除

```
curl -XGET 'http://192.168.4.14:9200/tarena/teacher/1'
curl -XDELETE 'http://192.168.4.14:9200/tarena/teacher/1'
curl -XDELETE 'http://192.168.4.14:9200/tarena'
```



案例5：插入，增加，删除查询数据

1. 使用curl命令连接使用ES数据库
2. 使用PUT方法增加数据
3. 使用POST修改数据
4. 使用GET查询数据
5. 使用DELETE删除数据



Kibana安装

Kibana安装

Kibana

Kibana安装与配置

kibana

Kibana安装与配置

- kibana是什么
 - 数据可视化平台工具
- 特点：
 - 灵活的分析 and 可视化平台
 - 实时总结流量和数据的图表
 - 为不同的用户显示直观的界面
 - 即时分享和嵌入的仪表板



Kibana安装与配置（续1）

- kibana安装
 - kibana 的安装非常简单，我们使用 rpm 方式安装
`rpm -ivh kibana-4.5.2-1.x86_64.rpm`
 - kibana 默认安装在 /opt/kibana 下面，配置文件在 /opt/kibana/config/kibana.yml
 - 我们只需要修改少量的配置就可以启动



Kibana安装与配置（续2）

- kibana.yml的配置
 - server.port: 5601
 - server.host: "0.0.0.0"
 - elasticsearch.url: "http://192.168.4.13:9200"
 - kibana.index: ".kibana"
 - kibana.defaultAppId: "discover"
 - elasticsearch.pingTimeout: 1500
 - elasticsearch.requestTimeout: 30000
 - elasticsearch.startupTimeout: 5000



Kibana安装与配置（续3）

- kibana.yml的配置
 - 除elasticsearch.url需要配置为我们ES集群的地址之外，其他保持默认值
 - 设置开机启动
`systemctl enable kibana`
 - 启动服务
`systemctl start kibana`
 - web访问kibana
`http://192.168.4.20:5601/`



案例6：安装Kibana

1. 安装Kibana
2. 配置启动服务查看5601端口是否正常
3. 通过web页面访问Kibana



总结和答疑
