**Morning：**

1. JSON、YAML与Jinja2

**-JSON**

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。

数据由”,”号分隔；保存对象(键值对)使用”{a:b}”；保存数组使用”[]”

{ "讲师":

[{"牛犇":"小逗逼", "负责阶段":"1"},

{"丁丁":"老逗逼", "负责阶段":"2"},

{"静静":"漂亮姐", "负责阶段":"3"},

{"李欣":"老司机", "负责阶段":"4"}

]

}

**-YAML**

一个可读性高，用来表达数据序列的格式(.yml)。

数组使用”- ”(-空格)，键值对使用”: ”(:空格)；同一层缩进对齐(空格)；

不同层级缩进至少2空格

"讲师":

-

"牛犇": "小逗比"

"阶段": 1

-

"丁丁": "老逗比"

"阶段": 2

-

"静静": "漂亮姐"

"阶段": 3

-

"李欣": "老司机"

"阶段": 4

**-Jinja2模板**

基于Python的模板引擎，包含变量和表达式两部分，支持多种语法。

表达式使用"{{ }}"；控制语句使用"{% %}"；注释使用"{# #}"

{{varname}} #调用变量

{{2+3}} #计算

{{5 in [3,4,5]}} #判断

{% if name == '小逗逼' %} #控制语句

讲故事,吹牛B

{% elif name == '老逗逼' %}

黑丝(82年的)

{% else %}

沉迷学习,无法自拔

{% endif %}

过滤器：类似shell的管道，查询以下在线文档

http://docs.jinkan.org/docs/jinja2/templates.html

1. ELK概述

**-What’s ELK**

ELK用于日志数据的集中查询和管理，包括以下组件：

Elasticsearch：负责日志检索与存储；（类比LNMP的M）

Logstash：负责日志的收集与分析、处理；（类比LNMP的P）

Kibana：负责日志的可视化；（类比LNMP的N）

**-Elasticsearch部分**

提供了一个分布式多用户能力的全文搜索引擎（数据库）。

Node：一个ES服务器节点

Cluster：多个Node组成的集群

Document：可被搜索的基础信息单元(行)

Index：拥有相似特征的文档集合(库)

Type：一个索引中可以定义一种或多种类型(表)

Fields：ES的最小单位(列)

Shards：索引的分片

Replicas：索引的拷贝

**-Logstash部分**

一个数据采集、加工处理及传输的工具。

Input{ } ==> filter{ } ==> output{ }

数据采集 加工处理 输出到ES

**-Kibana部分**

通过读取ES的数据，以web页面展示图形。

1. ES集群安装

**步骤1：准备工作(单台)**

1. 准备虚拟机

2核CPU、1.5G内存、20G硬盘

2）设置IP、主机名

[root@es1 ~]# cat /etc/hosts

...

192.168.1.11 es1

3）安装OpenJDK 1.8以上版本

[root@es1 ~]# yum -y install java-1.8.0-openjdk

**步骤2：安装ES、修改配置(单台)**

[root@es1 yum.repos.d]# cat es.repo #搭建yum源

[Elasticsearch]

...

baseurl=ftp://192.168.1.254/es\_yum

[root@es1 ~]# yum -y install elasticsearch #装包

[root@es1 ~]# vim /etc/elasticsearch/elasticsearch.yml

network.host: 192.168.1.11

**步骤3：启动服务、验证**

[root@es1 ~]# systemctl start elasticsearch

[root@es1 ~]# systemctl enable elasticsearch

[root@es1 ~]# netstat -anputl | grep java

9200、9300端口被监听

[root@es1 ~]# curl 192.168.1.11:9200

{

"name" : "Smart Alec",

"cluster\_name" : "elasticsearch"

...

}

**步骤4：多准备4台node(集群)**

1. 分别配置主机名、IP地址
2. 分别配置/ets/hosts及yum

[root@es1 ~]# for i in es{2..5}

> do

> scp /etc/hosts $i:/etc/

> scp /etc/yum.repos.d/

> scp /etc/yum.repos.d/es.repo $i:/etc/yum.repos.d/

> done

1. 分别安装openjdk、elasticsearch

[root@es1 ~]# for i in es{2..5}

> do

> ssh $i yum -y install java-1.8.0-openjdk elasticsearch

> done

1. 分别修改配置文件(集群模式)

[root@es1 ~]# vim /etc/elasticsearch/elasticsearch.yml

cluster.name: nsd1808 #集群名(相同则为同一集群)

node.name: es1 #本机主机名

network.host: 192.168.1.11 #本机IP

discovery.zen.ping.unicast.hosts: ["es1", "es2", "es3"]

#声明集群节点(用于发现其他成员，必须保证其中一个节点green)

[root@es1 ~]# systemctl {start|enable} elasticsearch

5）验证集群

[root@es1 ~]# for i in es{1..5}; do

> curl http://$i:9200/\_cluster/health?pretty

> done

{

"cluster\_name" : "nsd1808",

"status" : "green",

"number\_of\_nodes" : 5,

...

}

**Afternoon：**

1. HTTP协议与Curl命令

**-http组成及请求方法**

请求行(①～③)、消息报头(④)、请求正文(⑤)



ES常用http方法：PUT(增)、DELETE(删)、POST(改)、GET(查)

**-curl命令**

利用URL规则在命令行下工作的文件传输工具(支持多种协议)。

常见参数： -A 修改请求agent (修改HTTP\_USER\_AGENT伪装浏览器)

-X设置请求方法

-i 显示响应头部信息+正文(相当于get)

-I 只显示响应头部信息(相当于head)

-v 显示请求及响应的所有信息(>开头为请求、<为响应)

-s 关闭统计信息的显示，用于curl -s url > /dev/null

测试： curl -A "Mozilla/5.0" http://118.144.89.240/info.php

curl -X POST http://118.144.89.240/info.php

curl -i http://118.144.89.240/info.php

HTTP/1.1 200 OK

Server: nginx 头部信息

...

<pre>

[ REQUEST\_METHOD] ==> GET

[ REMOTE\_ADDR] ==> 113.67.159.74 正文信息

[HTTP\_USER\_AGENT] ==> curl/7.29.0

2、ES插件的使用

**-安装ES插件**

[root@es1 ~]# cd /usr/share/elasticsearch/bin/

~]# ./plugin install ftp://192.168.1.254/es\_yum/el..-head-ma..zip

~]# ./plugin install ftp://192.168.1.254/es\_yum/el..-kopf-mas..zip

~]# ./plugin install ftp://192.168.1.254/es\_yum/bigdesk-mas..zip

~]# ./plugin list

Tips： -plugin工具用于管理/安装插件

**head**插件用于展现ES集群的拓扑结构

ospf插件用于管理集群，提供操作集群的API

bigdesk插件用于监控集群

**-访问ES插件**

http://192.168.1.11:9200/\_plugin/head

http://192.168.1.11:9200/\_plugin/kopf

http://192.168.1.11:9200/\_plugin/bigdesk

3、拓展插件RESTful API调用

**1）\_cat查询**

curl http://192.168.1.13:9200/\_cat/

/\_cat/shards #查看分片信息

/\_cat/nodes #查看节点状态信息

/\_cat/indices #查看索引信息

curl http://192.168.1.13:9200/\_cat/nodes?help

curl http://192.168.1.13:9200/\_cat/nodes?v

#?v显示详细信息 ?help显示帮助信息

1. **插入数据(PUT)**

##创建索引(分片数量5,副本数量1)##

[root@es1 ~]# curl -XPUT 192.168.1.11:9200/nsd1808 -d '{

“settings”: {

“index”: {

“number\_of\_shards”: 5,

“number\_of\_replicas”: 1 }

}

}'

##增加文档##

[root@es1 ~]# curl -XPUT http://192.168.1.11:9200/nsd1808/js/1 -d '{

"姓名": "牛牛",

"爱好": "LOL",

"年龄": 20,

"阶段": 1

}'

[root@es1 ~]# curl -XPUT http://192.168.1.11:9200/nsd1808/js/2 -d '{

"姓名": "丁丁",

"爱好": "吃喝",

"年龄": 30,

"阶段": 2

}'

#nsd1808/js/2分别为：索引名/类型名/行号

1. **修改、查询、删除数据(POST/GET/DELETE)**

##修改文档内容##

[ ~]# curl -XPOST http://192.168.1.11:9200/nsd1808/js/2/\_update \

> -d '{

> "doc":{ "年龄": 50 }

> }'

#关键字\_update，json格式的key为”doc”

##查询与删除##

[root@es1 ~]# curl -XGET http://192.168.1.11:9200/nsd1808/js/2

[root@es1 ~]# curl -XDELETE http://192.168.1.11:9200/nsd1808/js/2

#删除单行

[root@es1 ~]# curl -XDELETE http://192.168.1.11:9200/nsd1808

#删除索引(库)

[root@es1 ~]# curl -XDELETE http://192.168.1.11:9200/\*

#删除所有索引(库)