**Logstash**

**Logstash是什么**

**• 是一个数据采集、加工处理以及传输的工具**

**• 特点**

**– 所有类型的数据集中处理**

**– 不同模式和格式数据的正常化**

**– 自定义日志格式的迅速扩展**

**– 为自定义数据源轻松添加插件**

**• Logstash安装**

**– Logstash依赖Java环境,需要安装java-1.8.0-openjdk**

**– Logstash没有默认的配置文件,需要手动配置**

**– Logstash安装在/opt/logstash目录下**

**# rpm -ivh logstash-2.3.4-1.noarch.rpm**

**[root@logstash ~]# cd /opt/logstash/bin**

**[root@logstash bin]# ./logstash**

**No command given**

**Usage: logstash <command> [command args]**

**Run a command with the --help flag to see the arguments.**

**For example: logstash agent --help**

**Available commands:**

**agent - runs the logstash agent**

**version - emits version info about this logstash**

**[root@logstash bin]# touch /etc/logstash/logstash.conf**

**[root@logstash ~]# cat /etc/hosts**

**192.168.1.61 es1**

**192.168.1.62 es2**

**192.168.1.63 es3**

**192.168.1.64 es4**

**192.168.1.65 es5**

**192.168.1.66 kibana**

**192.168.1.67 logstash**

**1 设置/etc/hosts logstash 要访问es机群**

**2 安装依赖包java-1.8.0-openjdk**

**3 安装包 logstash**

**4 创建空的配置文件 touch /etc/logstash/logstash.conf**

**• Logstash工作结构**

**{ 数据源 } ==>**

**input { } ==> （如何采集数据的）**

**filter { } ==> （如何处理数据）**

**output { } ==> （把数据发送到哪里去）**

**{ ES }**

**• Logstash里面的类型**

**– 布尔值类型: ssl\_enable => true**

**– 字节类型: bytes => "1MiB"**

**– 字符串类型: name => "xkops"**

**– 数值类型: port => 22**

**– 数组: match => ["datetime","UNIX"]**

**– 哈希: options => {k => "v",k2 => "v2"}**

**– 编码解码: codec => "json"**

**– 路径: file\_path => "/tmp/filename"**

**– 注释: #**

**• Logstash条件判断**

**– 等于: ==**

**– 不等于: !=**

**– 小于: <**

**– 大于: >**

**– 小于等于: <=**

**– 大于等于: >=**

**– 匹配正则: =~**

**– 不匹配正则: !~**

**– 包含: in**

**– 不包含: not in**

**– 与: and**

**– 或: or**

**– 非与: nand**

**– 非或: xor**

**– 复合表达式: ()**

**– 取反符合: !()**

**[root@logstash bin]# /opt/logstash/bin/logstash-plugin 插件管理**

**• Logstash插件**

**– 上页的配置文件使用了logstash-input-stdin和logstash-output-stdout两个插件, Logstash还有filter和codec类插件,查看插件的方式是**

**# /opt/logstash/bin/logstash-plugin list**

**Codec类插件 编码插件**

**Filter类插件 过滤插件 处理数据**

**Input插件 输入插件**

**Output插件 输出插件**

**[root@logstash bin]# cat /etc/logstash/logstash.conf**

**input {**

**stdin{}**

**}**

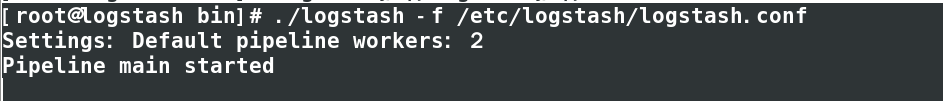
**filter{ }**

**output{**

**stdout{}**

**}**

**启动验证**

****

**– 插件及文档地址**

**<https://github.com/logstash-plugins>**

**• codec类插件**

**– 常用的插件:plain、json、json\_lines、rubydebug、**

**multiline等**

**– 使用刚刚的例子,这次输入json数据**

**– 设置输入源的codec是json,在输入的时候选择**

**rubydebug**

**[root@logstash bin]# cat /etc/logstash/logstash.conf**

**input {**

**stdin{ codec => "json" } 输入json格式**

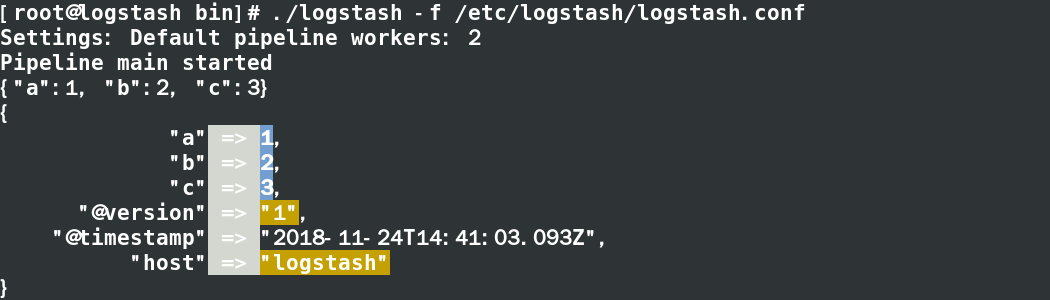
**}**

**filter{ }**

**output{**

**stdout{ codec => "rubydebug" } 输出格式调整 列显示**

**}**

****

**监控文件 使用file插件**

**• 练习input file插件**

**file{**

**start\_position => "beginning"**

**sincedb\_path => "/var/lib/logstash/sincedb-access"**

**path => [“/tmp/a.log”, “/tmp/b.log”]**

**type => 'filelog'**

**}**

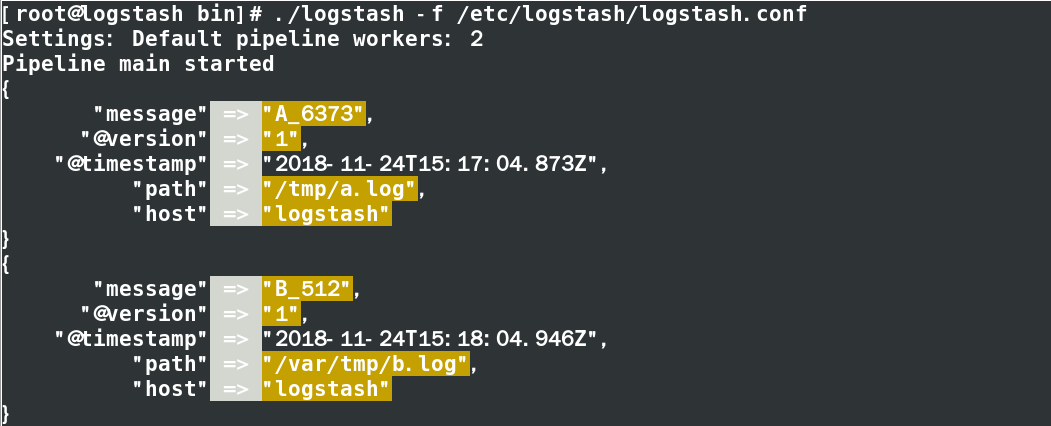
**– sincedb\_path记录读取文件的位置**

**– start\_position配置第一次读取文件从什么地方开始 默认行为**

****

**[root@logstash ~]# echo A\_${RANDOM} >> /tmp/a.log**

**[root@logstash ~]# echo B\_${RANDOM} >> /var/tmp/b.log**

****

**• 练习input tcp和udp插件**

**tcp{ tcp服务器端口 随机高端端口**

**Mode => “server” 服务器本身 client 指的是远端的服务器**

**host => "0.0.0.0" 本机的地址 目标地址**

**port => 8888 本机的ip地址 目标端口**

**type => "tcplog"**

**}**

**udp{ udp 默认server 只有服务器端**

**port => 8888**

**type => "udplog"**

**}**

**[root@web1 ~]# function sendtcp(){ tcp 脚本**

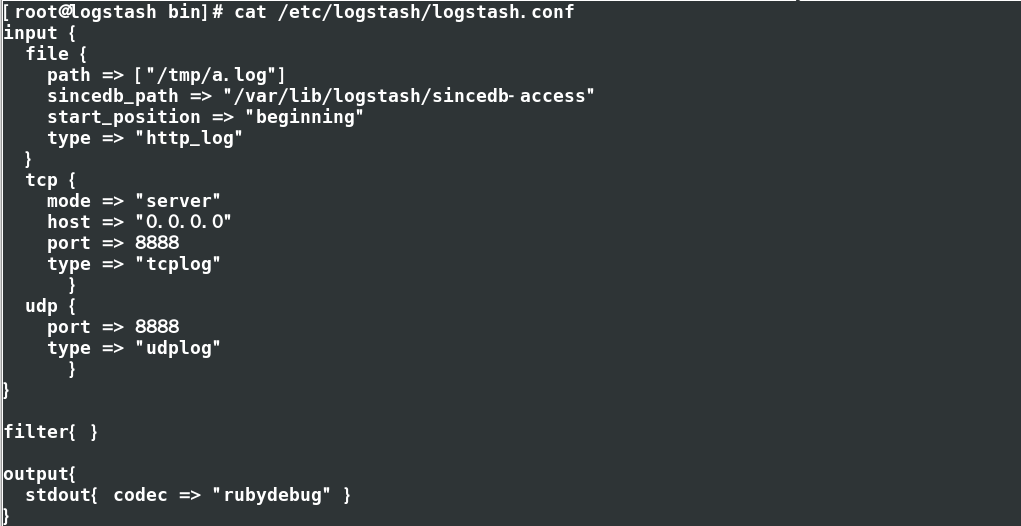
**> exec 9<>/dev/tcp/192.168.1.67/8888**

**> echo "$@" >&9**

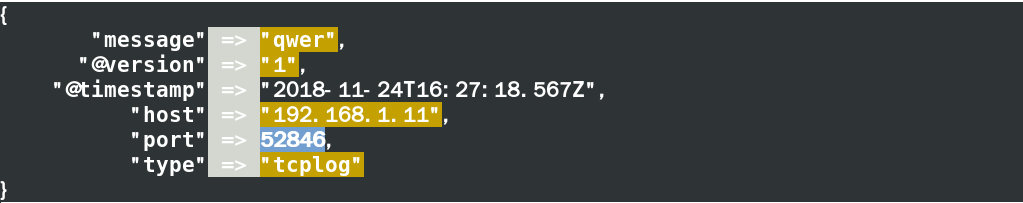
**> exec 9<&-**

**> }**

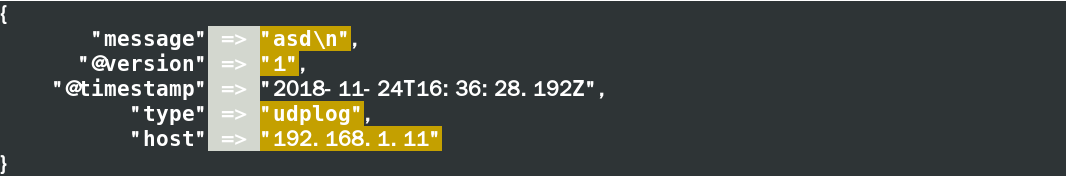
**[root@web1 ~]# sendtcp qwer**

****

**[root@logstash bin]# ./logstash -f /etc/logstash/logstash.conf**

****

**[root@web1 ~]# sendudp asd**

****

**• syslog插件练习**

**syslog{**

**host => "0.0.0.0"**

**port => 514 syslog默认端口号**

**type => "syslog"**

**}**

**– rsyslog.conf配置向远程发送数据**

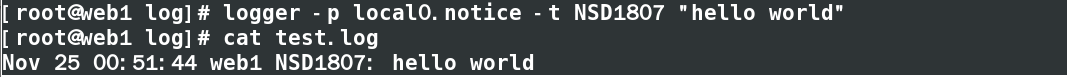
**local0.info**

**@@192.168.4.10:514**

**– 写syslog,查看状态**

**logger -p local0.info -t test\_logstash 'test message'**

**写系统日志**

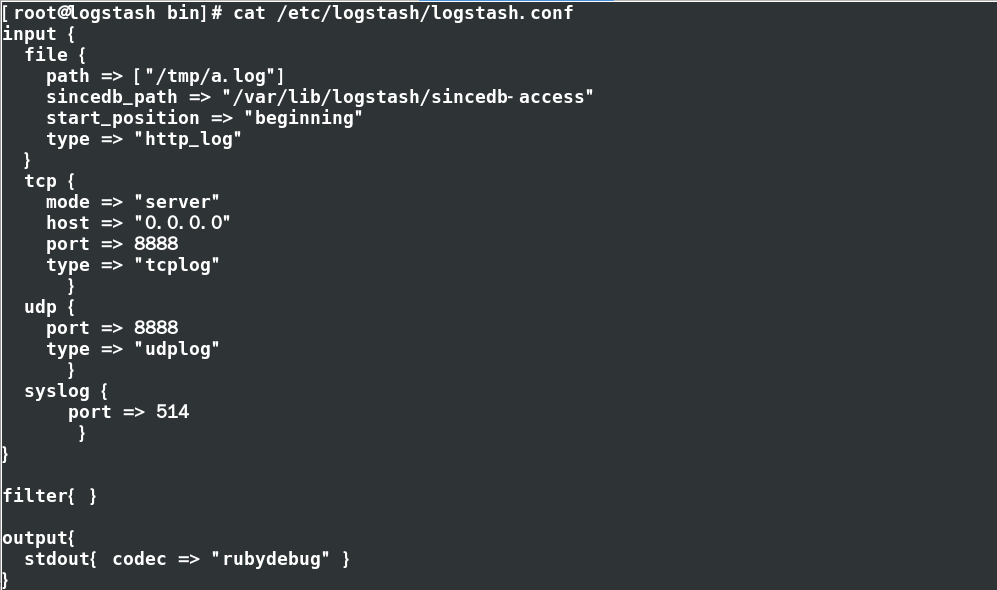
****

**[root@web1 log]# vim /etc/rsyslog.conf**

**authpriv.\* @@192.168.1.67:514 安全监控 如ssh连接本机**

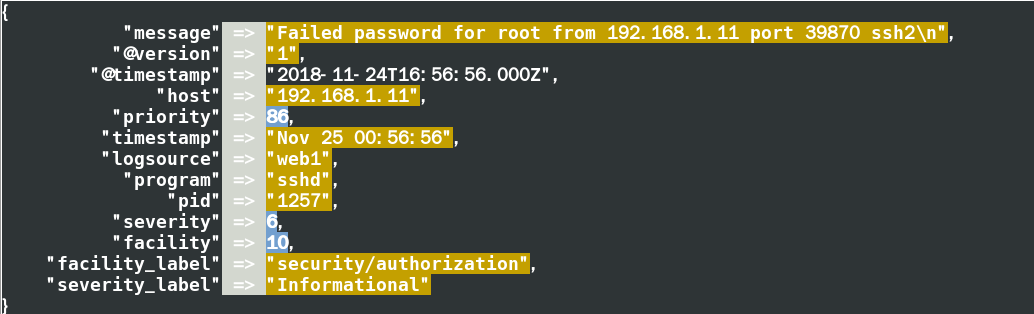
**local0.=notice @@192.168.1.67:514 日志类型等级**

**\*.\* @@192.168.1.67:514 指定日志存放于监控端**

****

**[root@logstash bin]# ./logstash -f /etc/logstash/logstash.conf**

**记录失败ssh连接失败 错误原因 连接的ip地址 以及ssh登陆成功也会记录**

****

**金步国翻译各种手册**

**filter grok插件**

**– 解析各种非结构化的日志数据插件**

**– grok使用正则表达式把飞结构化的数据结构化**

**– 在分组匹配,正则表达式需要根据具体数据结构编写**

**– 虽然编写困难,但适用性极广**

**– 几乎可以应用于各类数据**

**grok{**

**match => ["message", "%{IP:ip}, (?<key>reg) " ]**

**}**

**• grok正则分组匹配**

**– 匹配ip时间戳和请求方法**

**"(?<ip>(\d+\.){3}\d+) \S+ \S+**

**(?<time>.\*\])\s+\"(?<method>[A-Z]+)"]**

**– 使用正则宏**

**%{IPORHOST:clientip} %{HTTPDUSER:ident} %{USER:auth}**

**\[%{HTTPDATE:timestamp}\] \"%{WORD:verb}**

**– 最终版本**

**%{COMMONAPACHELOG} \"(?<referer>[^\"]+)\"**

**\"(?<UA>[^\"]+)\"**

**• output ES插件**

**if [type] == "filelog"{**

**elasticsearch {**

**hosts => ["192.168.4.15:9200"]**

**index => "weblog"**

**flush\_size => 2000**

**idle\_flush\_time => 10**

**}}**

**– 调试成功后,把数据写入ES集群**

**• input filebeats插件**

**beats {**

**port => 5044**

**codec => "json"**

**}**

**– 这个插件主要用来接收beats类软件发送过来的数据,**

**由于logstash依赖JAVA环境,而且占用资源非常大,**

**因此会使用更轻量的filebeat替代**

**filebeat安装配置**

**• filebeat安装与配置**

**– 使用rpm安装filebeat**

**# rpm -ivh filebeat-1.2.3-x86\_64.rpm**

**– 修改配置文件/etc/filebeat/filebeat.yml**

**– 设置开机运行**

**# systemctl enable filebeat**

**– 开启服务**

**# systemctl start filebeat**

**修改配置文件/etc/filebeat/filebeat.yml**

**paths:**

**- /root/logs.jsonl**

**document\_type: weblog**

**... ...**

**paths:**

**- /root/accounts.json**

**document\_type: account**

**output:**

**#elasticsearch**

**logstash:**

**hosts: ["192.168.1.67:5044"]**