甲方安全看日志消息在ELK中的流转

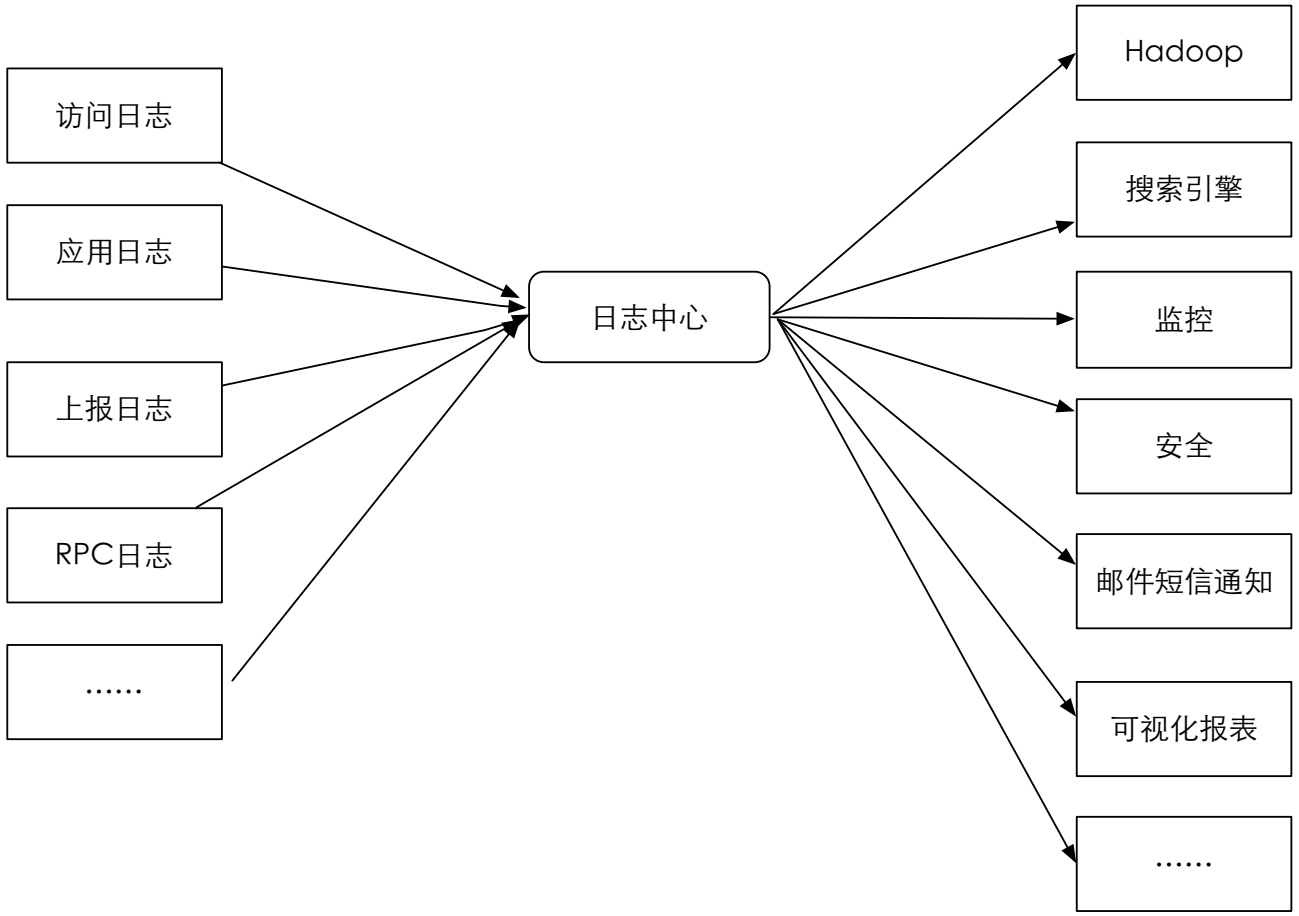
李天爽@闻心科技

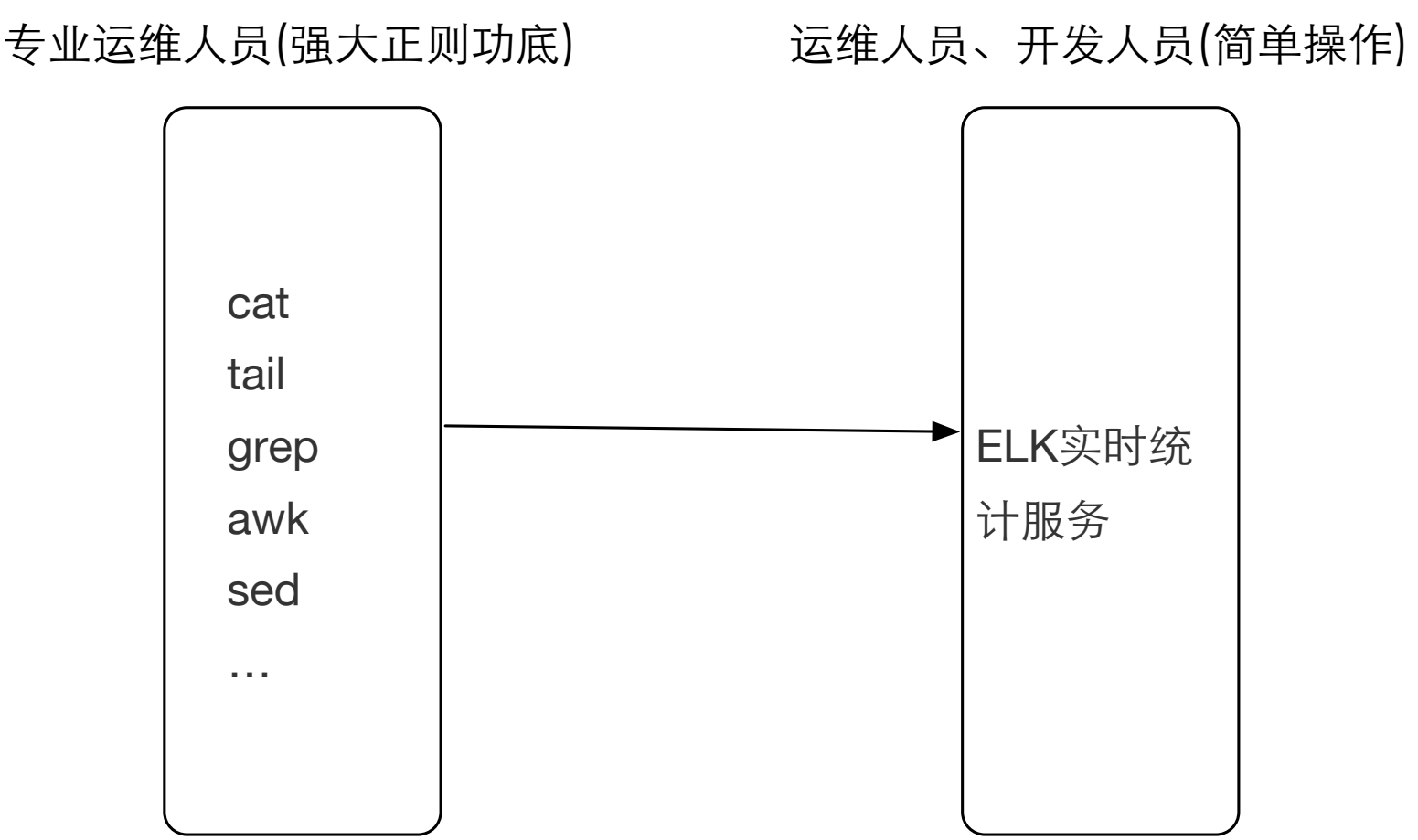
tianshuang.li@wen-xintech.com

微信: timesgarden0

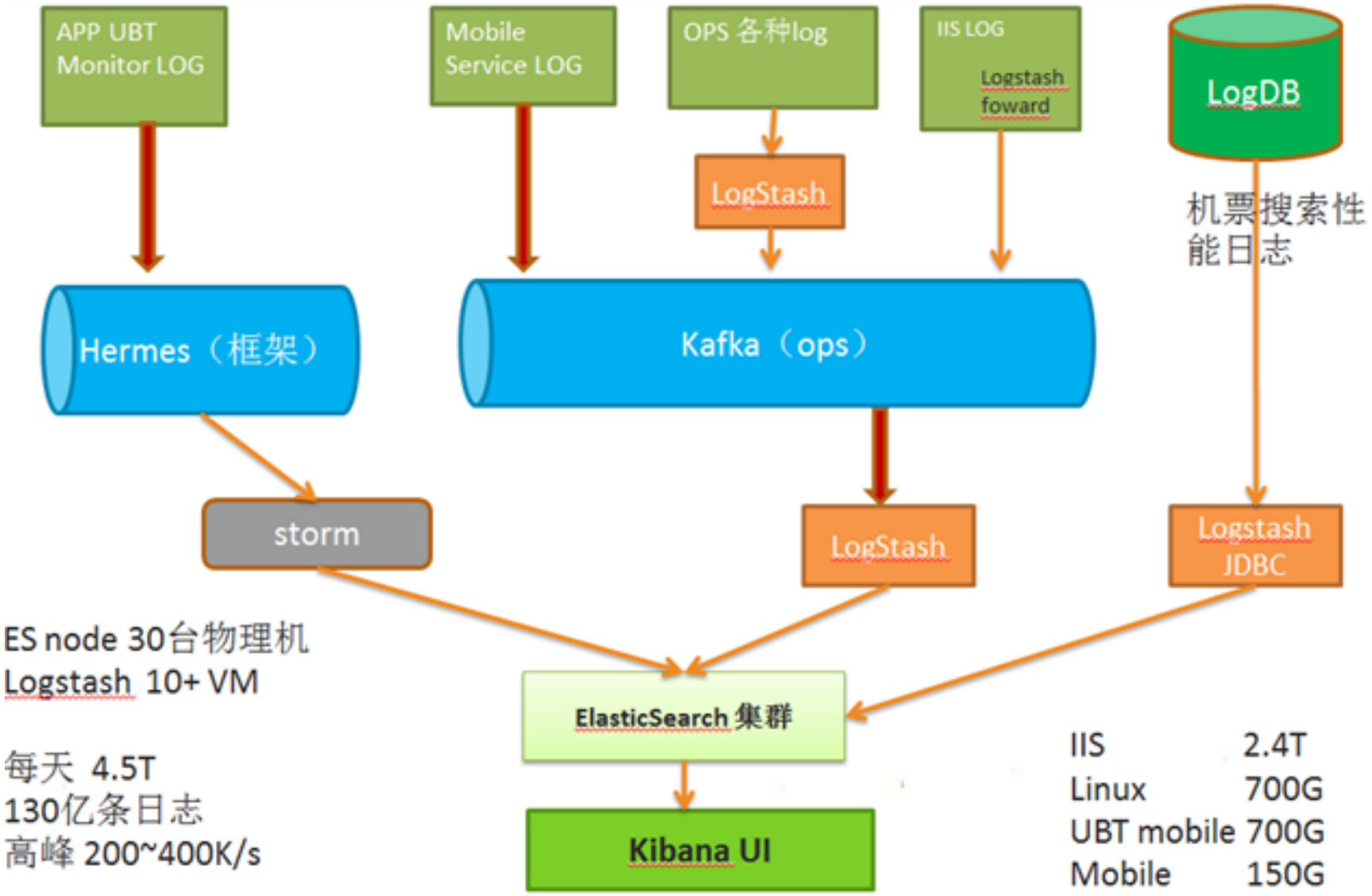
目录

日志用途




专业运维人员(强大正则功底)
运维人员、开发人员(简单操作)
cat
tail
grep
awk
sed
…
ELK实时统计服务

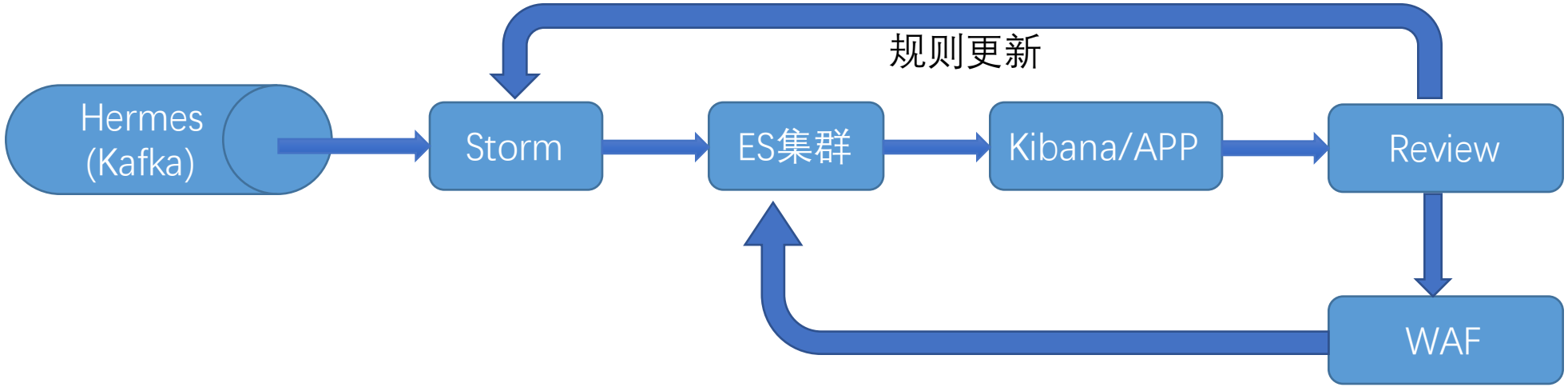
日志平台

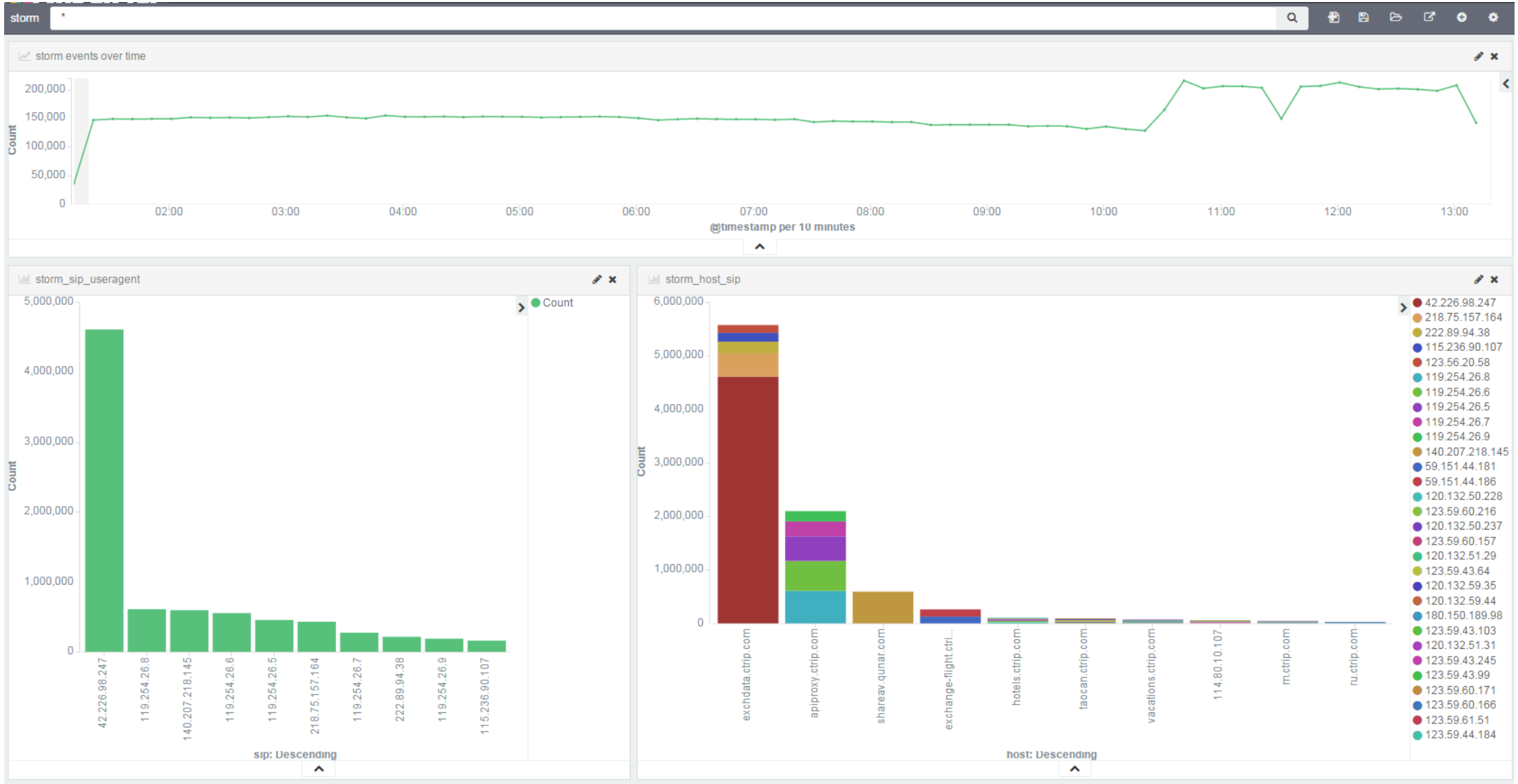

APP UBT
Monitor LOG
Mobile
Service LOG
OPS 各种log
IIS LOG
Logstash
foward
LogDB
LogStash
机票搜索性
能日志
Hermes（框架）
Kafka（ops）
storm
LogStash
Logstash
JDBC
ES node 30台物理机
Logstash 10+ VM
每天 4.5T
130亿条日志
高峰 200~400K/s
ElasticSearch 集群
Kibana UI
IIS 2.4T
Linux 700G
UBT mobile 700G
Mobile 150G

Kafka vs Redis

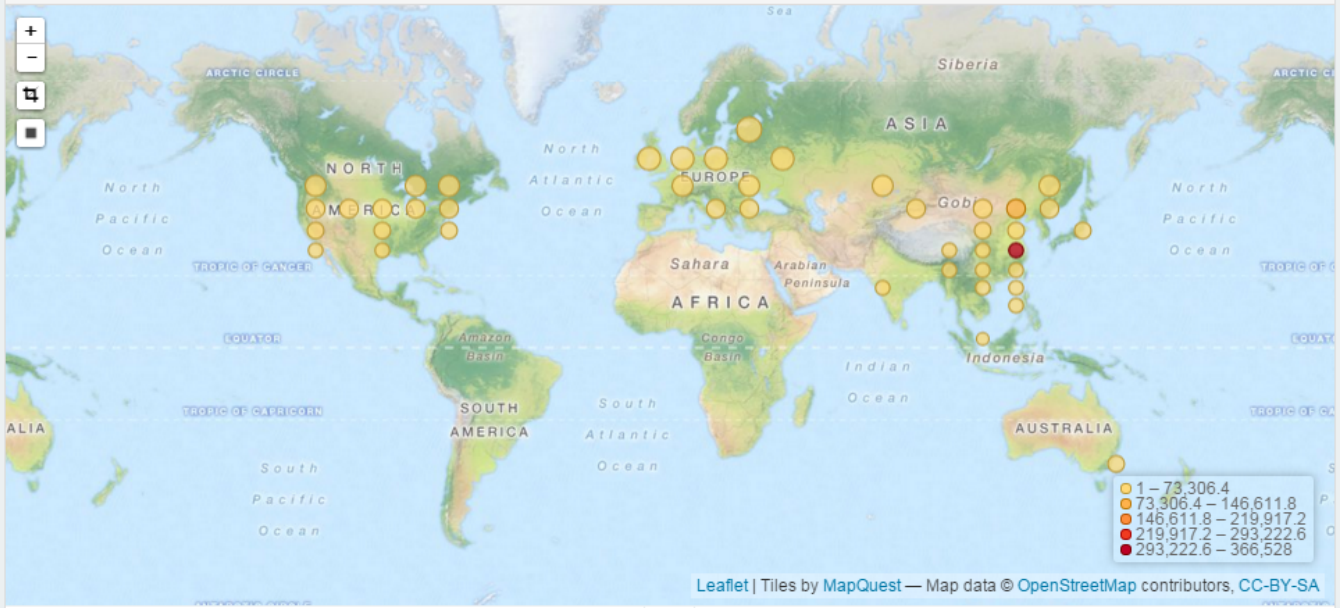
- 可靠性
- 消息堆积能力
- 日志领域成熟度
- 其他？

安全事件在ELK中的生命周期

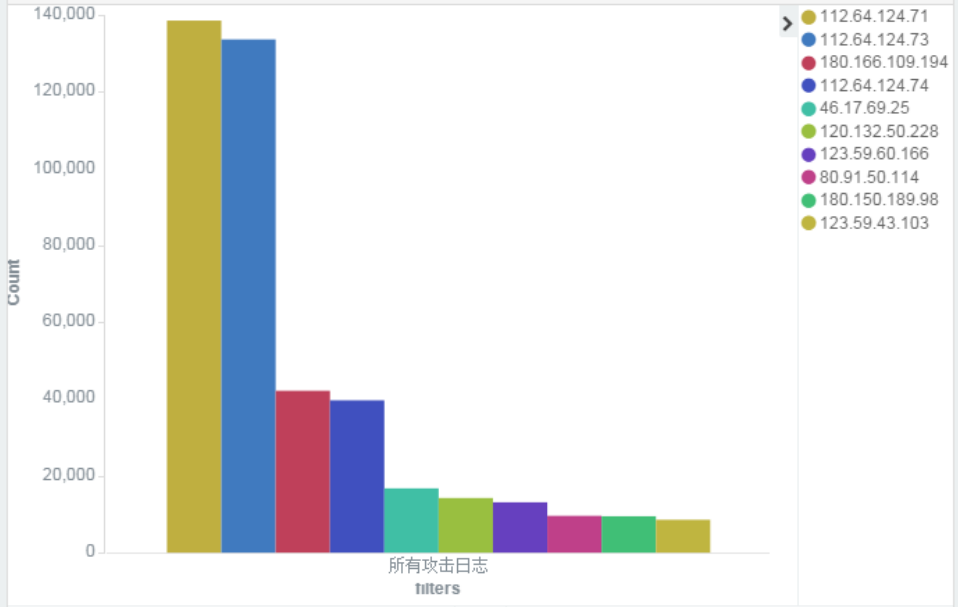



storm
*
storm events over time
200,000
150,000
100,000
50,000
0
Count
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
@timestamp per 10 minutes
storm_sip_useragent
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
Count
Count
42.226.98.247
119.254.26.8
140.207.218.145
119.254.26.6
119.254.26.5
218.75.157.164
119.254.26.7
222.89.94.38
119.254.26.9
115.236.90.107
sip: Descending
storm_host_sip
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
Count
exchdata.ctrip.com
apiproxy.ctrip.com
shareav.qunar.com
exchange-flight.ctri...
hotels.ctrip.com
taocan.ctrip.com
vacations.ctrip.com
114.80.10.107
m.ctrip.com
ru.ctrip.com
host: Descending
42.226.98.247
218.75.157.164
222.89.94.38
115.236.90.107
123.56.20.58
119.254.26.8
119.254.26.6
119.254.26.5
119.254.26.7
119.254.26.9
140.207.218.145
59.151.44.181
59.151.44.186
120.132.50.228
123.59.60.216
120.132.50.237
123.59.60.157
120.132.51.29
123.59.43.64
120.132.59.35
120.132.59.44
180.150.189.98
123.59.43.103
120.132.51.31
123.59.43.245
123.59.43.99
123.59.60.171
123.59.60.166
123.59.61.51
123.59.44.184

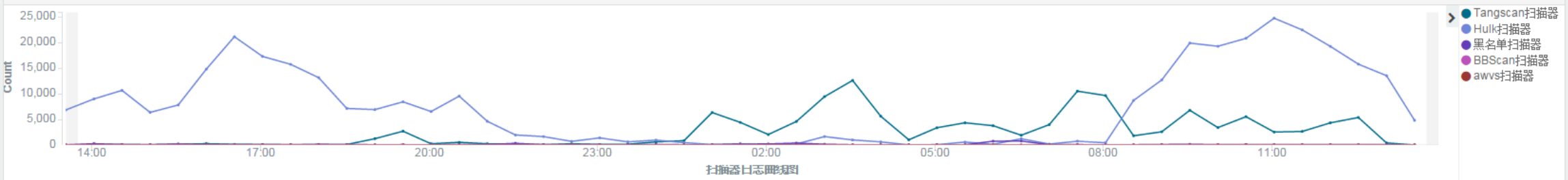
WAF-地图可视化



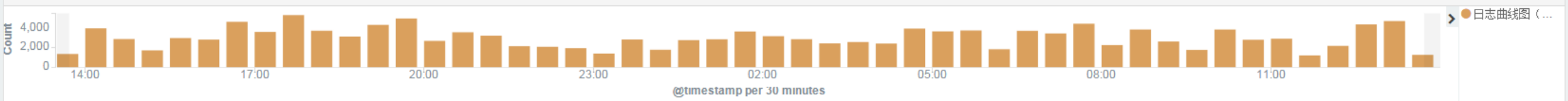
WAF-攻击源IPTop10



WAF-扫描器日志曲线图



WAF-日志柱状图（除扫描器）



历史攻击

攻击类型

4050	SQLi
1592	XSS
9	Other
6	APT IDS
2	账号

攻击省份 TOP10

2135	河南
1800	北京
460	湖北
366	上海
331	湖南
206	广东
185	浙江
44	四川
38	江苏
38	山东

攻击城市 TOP10

2135	新乡 (河南)
1800	北京 (北京)
459	武汉 (湖北)
366	上海 (上海)
329	常德 (湖南)
197	深圳 (广东)
141	杭州 (浙江)
33	德阳 (四川)
31	青岛 (山东)
25	镇江 (江苏)

攻击IP TOP10

1842	42.226.98.247 (河南-新乡)
454	58.19.56.220 (湖北-武汉)
350	140.207.218.145 (上海-上海)
329	218.75.157.164 (湖南-常德)
318	119.254.26.8 (北京-北京)
285	119.254.26.6 (北京-北京)
228	119.254.26.5 (北京-北京)
142	119.254.26.7 (北京-北京)
88	115.236.90.107 (浙江-杭州)
87	119.254.26.9 (北京-北京)

账户攻击

6	APT IDS
2	账号

WEB 攻击数据

3627	exchdata.ctrip.com
1068	apiproxy.ctrip.com
350	shareav.qunar.com
165	exchange-flight.ctrip.com
134	114.80.10.107
116	hotels.ctrip.com
32	cards.ctrip.com
27	ru.ctrip.com
24	m.ctrip.com
19	es.ctrip.com

实时攻击

时间	位置	IP	类型	域名
2016/6/12 下午1:13:35	湖南-常德	218.75.157.164	SQLi	exchdata.ctrip.com
2016/6/12 下午1:13:35	北京-北京	123.59.44.184	SQLi	hotels.ctrip.com
2016/6/12 下午1:13:35	湖北-武汉	58.19.56.220	SQLi	exchdata.ctrip.com
2016/6/12 下午1:13:36	湖南-常德	218.75.157.164	SQLi	exchdata.ctrip.com
2016/6/12 下午1:13:36	上海-上海	140.207.218.145	XSS	shareav.qunar.com

攻击类型

106	InfoLeak
72	SQLi
23	whitelist
6	RCE-0102
3	xss-0303

攻击省份 TOP10

146	Shanghai
15	Idaho
11	Ontario

攻击城市 TOP10

146	Shanghai (Shanghai)
15	Meridian (Idaho)
11	Sudbury (Ontario)

攻击IP TOP10

68	180.166.109.194 (Shanghai-Shanghai)
43	112.64.124.73 (Shanghai-Shanghai)
35	58.246.10.89 (Shanghai-Shanghai)
28	80.91.50.114 ()
15	66.232.80.170 (Idaho-Meridian)
11	216.223.125.196 (Ontario-Sudbury)
10	43.242.215.28 ()

WAF 拦截数据

64	openapi.ctrip.com
46	you.ctrip.com
41	flights.ctrip.com
24	huodong.ctrip.com
9	m.ctrip.com
7	vbooking.ctrip.com
6	piao.ctrip.com
6	meeting.ctrip.com
3	cruise.ctrip.com
2	contents.ctrip.com

实时攻击

时间	位置	IP	类型	域名
2016/6/12 下午1:37:44	China-Shanghai-Shanghai	112.64.124.73	InfoLeak	m.ctrip.com
2016/6/12 下午1:37:45	China-Shanghai-Shanghai	112.64.124.73	InfoLeak	m.ctrip.com
2016/6/12 下午1:37:45	China-Shanghai-Shanghai	58.246.10.89	SQLi	flights.ctrip.com
2016/6/12 下午1:37:45	China-Shanghai-Shanghai	180.166.109.194	InfoLeak	vbooking.ctrip.com
2016/6/12 下午1:37:46	China-Shanghai-Shanghai	180.166.109.194	InfoLeak	you.ctrip.com

ElasticSearch权限方案探讨

- Shield
- Search Guard
- Nginx http auth basic
- Nginx + Lua

Shield

- https://www.elastic.co/products/shield

- 收费官方插件
- 30天免费试用
- 30天后，Shield将会屏蔽cluster health, cluster stats, index stats这几个API，其余功能不受影响

Shield

- 用户认证
- 权限控制
- 集群节点认证与信道加密
  - Shield使用SSL/TLS加密相应端口(9300)，防止集群被未授权的机器监听或干扰
- IP 过滤
- 审计
  - Shield可以在ElasticSearch的日志中输出每次鉴权操作的详细信息，包括用户名，操作，操作是否被允许等等

Search Guard

- https://github.com/floragunncom/search-guard

- 免费开源，但文档欠缺，学习成本高
- 支持基于user和role进行index的权限管理（只读，还是可读可写等）
- 在ES的配置（elasticsearch.yml）中支持角色和权限定义
- 通过REST http方式把ACL策略（下页）提交到ES

```
{
  "acl": [{
    "__Comment__": "Default is to execute all filters",
    "filters_bypass": [],
    "filters_execute": []
  }, {
    "__Comment__": "This means that every requestor (regardless of the requestors hostname and username) which has the admin role can do anything",
    "roles": [
      "admin"
    ],
    "filters_bypass": [
      "*"
    ],
    "filters_execute": []
  }, {
    "__Comment__": "This means that for the user user1 on index test only the actionrequestfilter.readonly will be executed, no other",
    "users": [
      "user1"
    ],
    "indices": [
      "test"
    ],
    "filters_bypass": [],
    "filters_execute": [
      "actionrequestfilter.readonly"
    ]
  }, {
    "__Comment__": "kibana index has no strict",
    "roles": [
      "admin"
    ],
    "indices": [
      ".kibana"
    ],
    "filters_bypass": ["*"],
    "filters_execute": []
  }]
```

Nginx http auth basic

- https://kyup.com/tutorials/set-http-authentication-nginx/

- 配置简单
- 权限控制粗粒度，只能针对index/type做授权

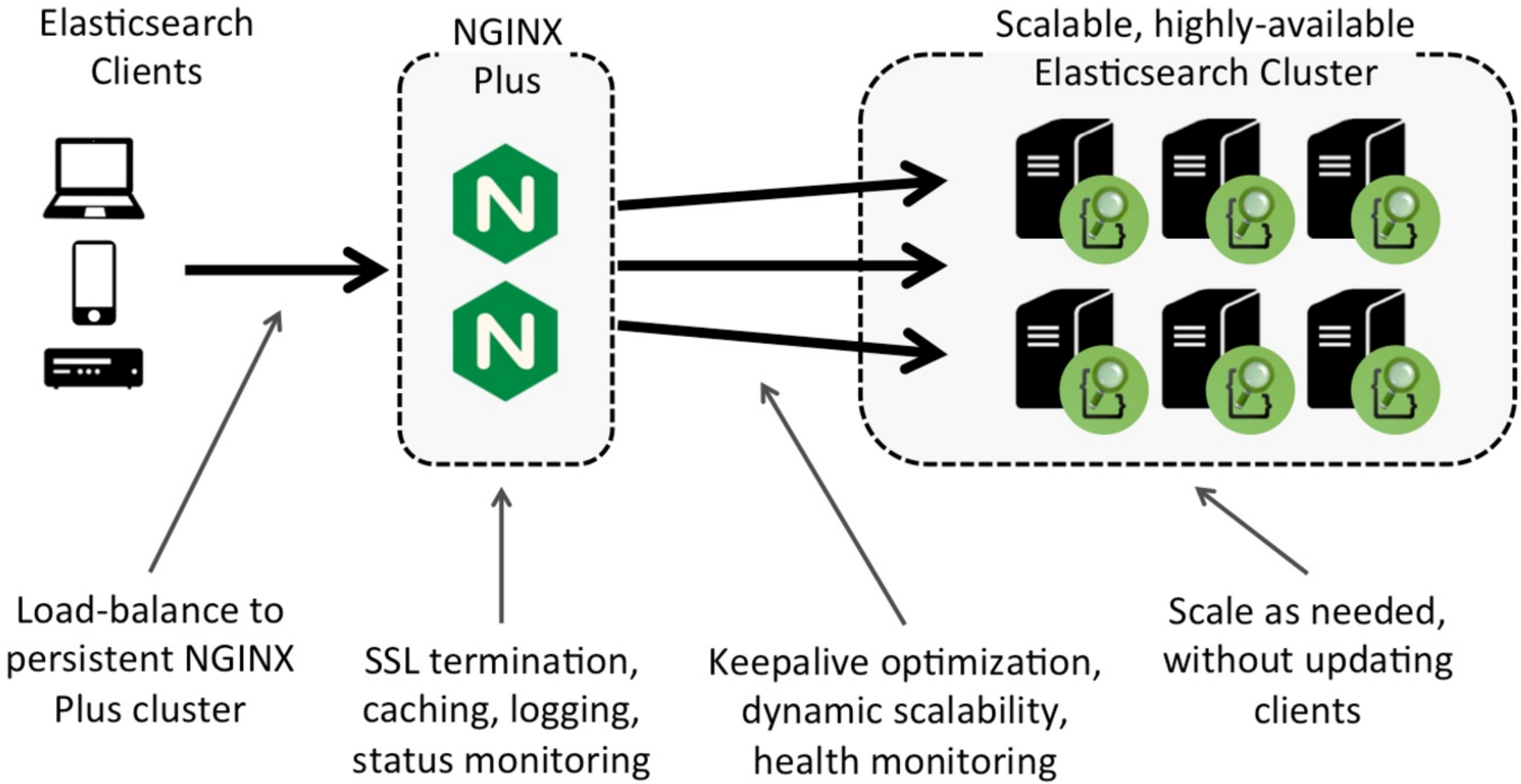
Nginx http auth basic

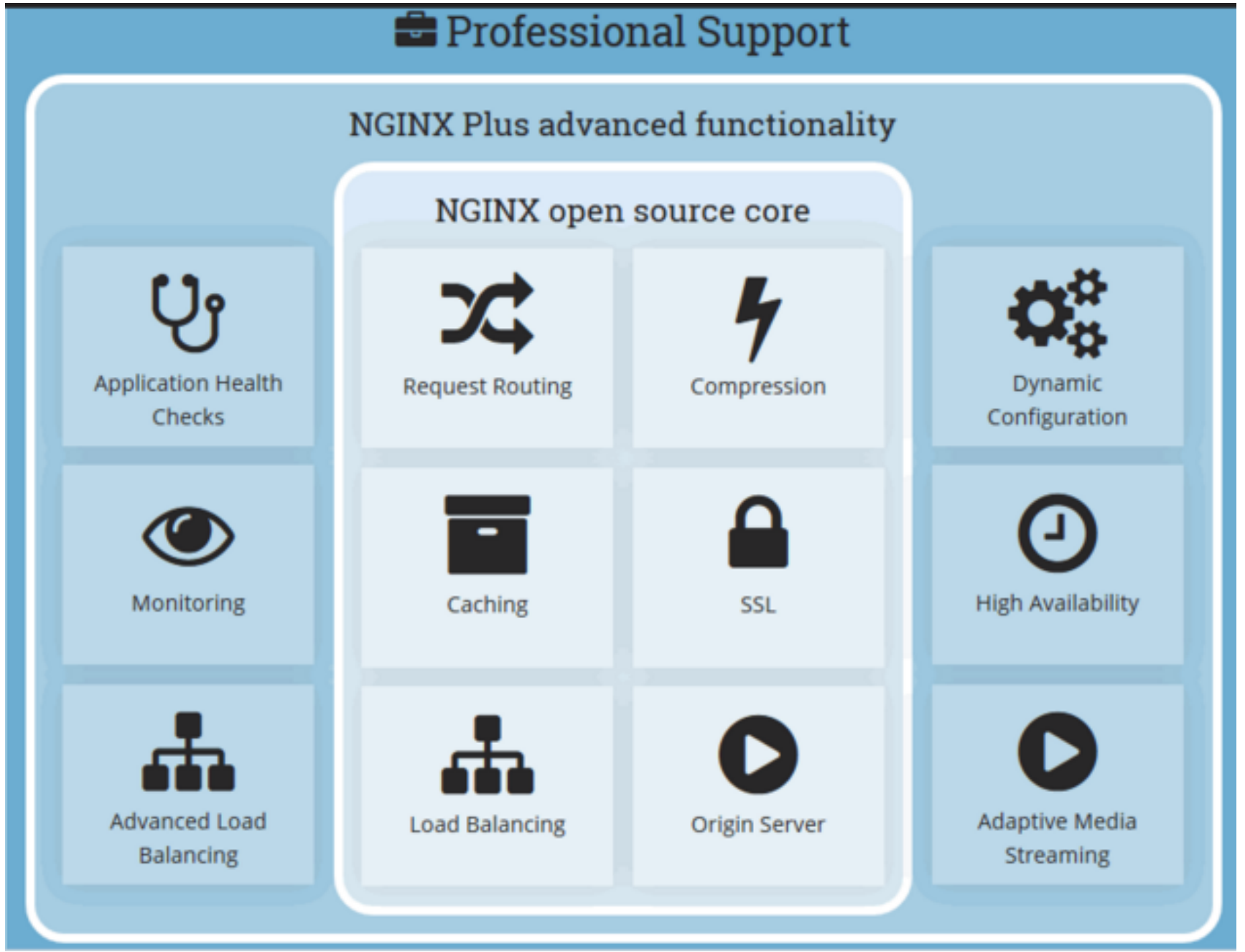
```
events {
  worker_connections  1024;
}

http {

  upstream elasticsearch {
      server 127.0.0.1:9200;
  }

  # Allow access to /_search and /_analyze for authenticated "users"
  #
  server {
      listen 8081;

      auth_basic           "Elasticsearch Users";
      auth_basic_user_file users;

      location / {
        return 403;
      }

      location ~* ^(/_search|/_analyze) {
        proxy_pass http://elasticsearch;
        proxy_redirect off;
      }
  }

  # Allow access to anything for authenticated "admins"
  #
  server {
      listen 8082;

      auth_basic           "Elasticsearch Admins";
      auth_basic_user_file admins;

      location / {
        proxy_pass http://elasticsearch;
        proxy_redirect off;
      }
  }

}
```

Why Nginx

- 记录每个API访问请求的日志（ElasticSearch本身不支持这个功能，只有slowLog和服务日志）
- 支持大量的客户端连接
- 负载均衡
- 缓存数据
- 提供主动健康检测（仅nginx plus）
- 安全验证
- 对特殊接口如"_shutdown"限制访问
- 带角色的访问控制（比如user角色拥有数据访问权限，admin角色拥有集群管控权限）

- Nginx的方法只能针对HTTP，如果走TCP（Java连接）的话就不work了


Elasticsearch Clients
NGINX Plus
Scalable, highly-available Elasticsearch Cluster
Load-balance to persistent NGINX Plus cluster
SSL termination, caching, logging, status monitoring
Keepalive optimization, dynamic scalability, health monitoring
Scale as needed, without updating clients


Professional Support
NGINX Plus advanced functionality
NGINX open source core
Application Health Checks
Request Routing
Compression
Dynamic Configuration
Monitoring
Caching
SSL
High Availability
Advanced Load Balancing
Load Balancing
Origin Server
Adaptive Media Streaming

Nginx+Lua

- https://www.elastic.co/blog/playing-http-tricks-nginx

- 用nginx lua的方式做auth扩展

```
$ curl -i -X HEAD 'http://localhost:8080'
# HTTP/1.1 401 Unauthorized
# ...

$ curl -i -X HEAD 'http://all:all@localhost:8080'
# HTTP/1.1 200 OK
# ...

$ curl -i -X GET 'http://all:all@localhost:8080'
# HTTP/1.1 403 Forbidden
# ...

$ curl -i -X GET 'http://user:user@localhost:8080'
# HTTP/1.1 200 OK
# ...

$ curl -i -X POST 'http://user:user@localhost:8080/myindex/mytype/1' -d '{"title" : "Test"}'
# HTTP/1.1 403 Forbidden
# ...

$ curl -i -X DELETE 'http://user:user@localhost:8080/myindex/'
# HTTP/1.1 403 Forbidden
# ...

$ curl -i -X POST 'http://admin:admin@localhost:8080/myindex/mytype/1' -d '{"title" : "Test"}'
# HTTP/1.1 200 OK
# ...

$ curl -i -X DELETE 'http://admin:admin@localhost:8080/myindex/'
# HTTP/1.1 200 OK
# ...
```

```
error_log logs/lua.log notice;

events {
  worker_connections 1024;
}

http {
  upstream elasticsearch {
    server 127.0.0.1:9200;
  }

  server {
    listen 8080;

    location / {
      auth_basic           "Protected Elasticsearch";
      auth_basic_user_file passwords;

      access_by_lua_file '../authorize.lua';

      proxy_pass http://elasticsearch;
      proxy_redirect off;
    }

  }
}
```

```
-- authorization rules

local restrictions = {
  all  = {
    ["^/$"]                           = { "HEAD" }
  },

  user = {
    ["^/$"]                           = { "GET" },
    ["^/?[^/]*/?[^/]*/_search"]        = { "GET", "POST" },
    ["^/?[^/]*/?[^/]*/_msearch"]       = { "GET", "POST" },
    ["^/?[^/]*/?[^/]*/_validate/query"] = { "GET", "POST" },
    ["/_aliases"]                     = { "GET" },
    ["/_cluster.*"]                   = { "GET" }
  },

  admin = {
    ["^/?[^/]*/?[^/]*/_bulk"]          = { "GET", "POST" },
    ["^/?[^/]*/?[^/]*/_refresh"]       = { "GET", "POST" },
    ["^/?[^/]*/?[^/]*/?[^/]*/_create"] = { "GET", "POST" },
    ["^/?[^/]*/?[^/]*/?[^/]*/_update"] = { "GET", "POST" },
    ["^/?[^/]*/?[^/]*/?.*"]            = { "GET", "POST", "PUT", "DELETE" },
    ["^/?[^/]*/?[^/]*$"]               = { "GET", "POST", "PUT", "DELETE" },
    ["/_aliases"]                     = { "GET", "POST" }
  }
}
```

```
-- get URL
local uri = ngx.var.uri

-- get method
local method = ngx.req.get_method()

local allowed  = false

for path, methods in pairs(restrictions[role]) do

  -- path matched rules?
  local p = string.match(uri, path)

  local m = nil

  -- method matched rules?
  for _, _method in pairs(methods) do
    m = m and m or string.match(method, _method)
  end

  if p and m then
    allowed = true
  end
end

if not allowed then
  ngx.header.content_type = 'text/plain'
  ngx.log(ngx.WARN, "Role ["..role.."] not allowed to access the resource ["..method.." "..uri.."]")
  ngx.status = 403
  ngx.say("403 Forbidden: You don't have access to this resource.")
  return ngx.exit(403)
end
```

Q & A?