
Guerrilla Blue Team
Logging and Alerting on a Shoestring Budget


Cboe
Global Markets

Disclaimer

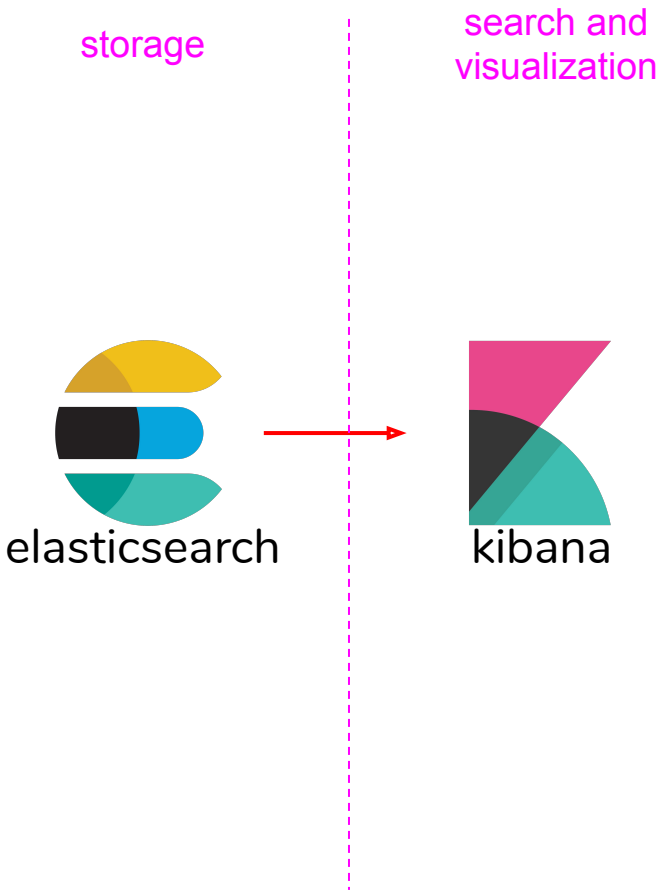
(or what Cboe lawyers would probably like us to say...if we'd actually taken the time to ask)

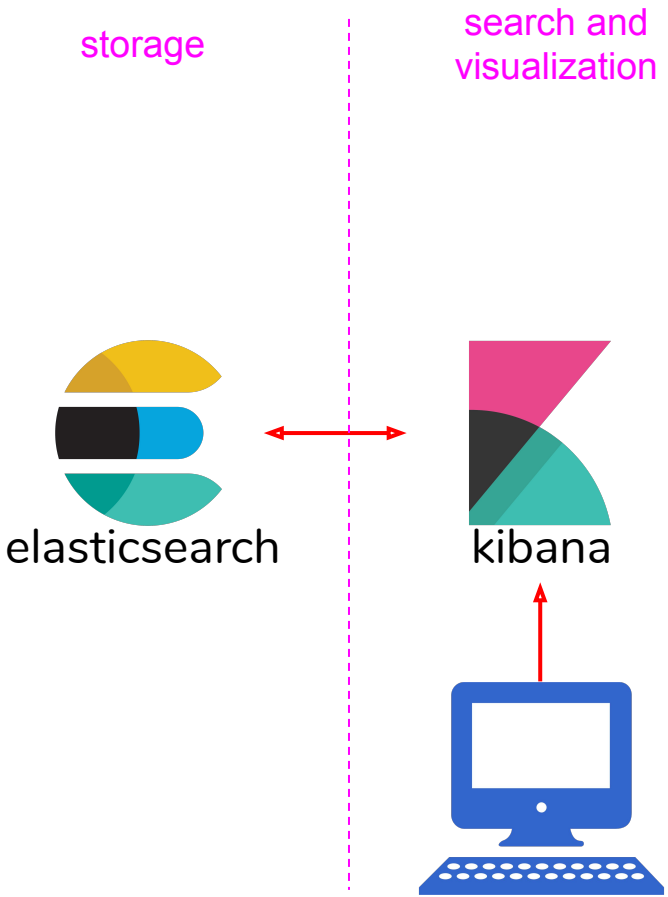
- The views and opinions expressed here are our own and do not necessarily represent those of our employer
- Our employer has absolutely nothing to do with anything related to this workshop content or material and does not warrant any of the advice, code or content in any way
- We're not speaking in the representation of our company

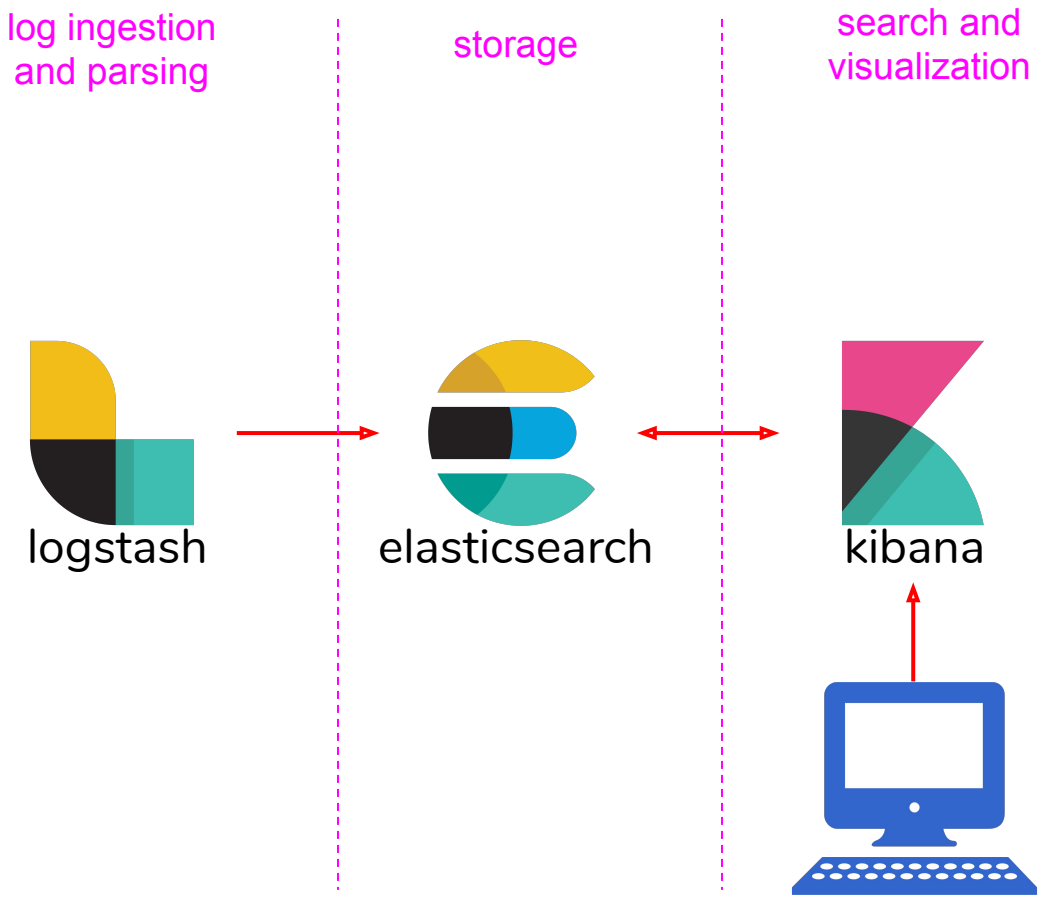
Workshop Outline

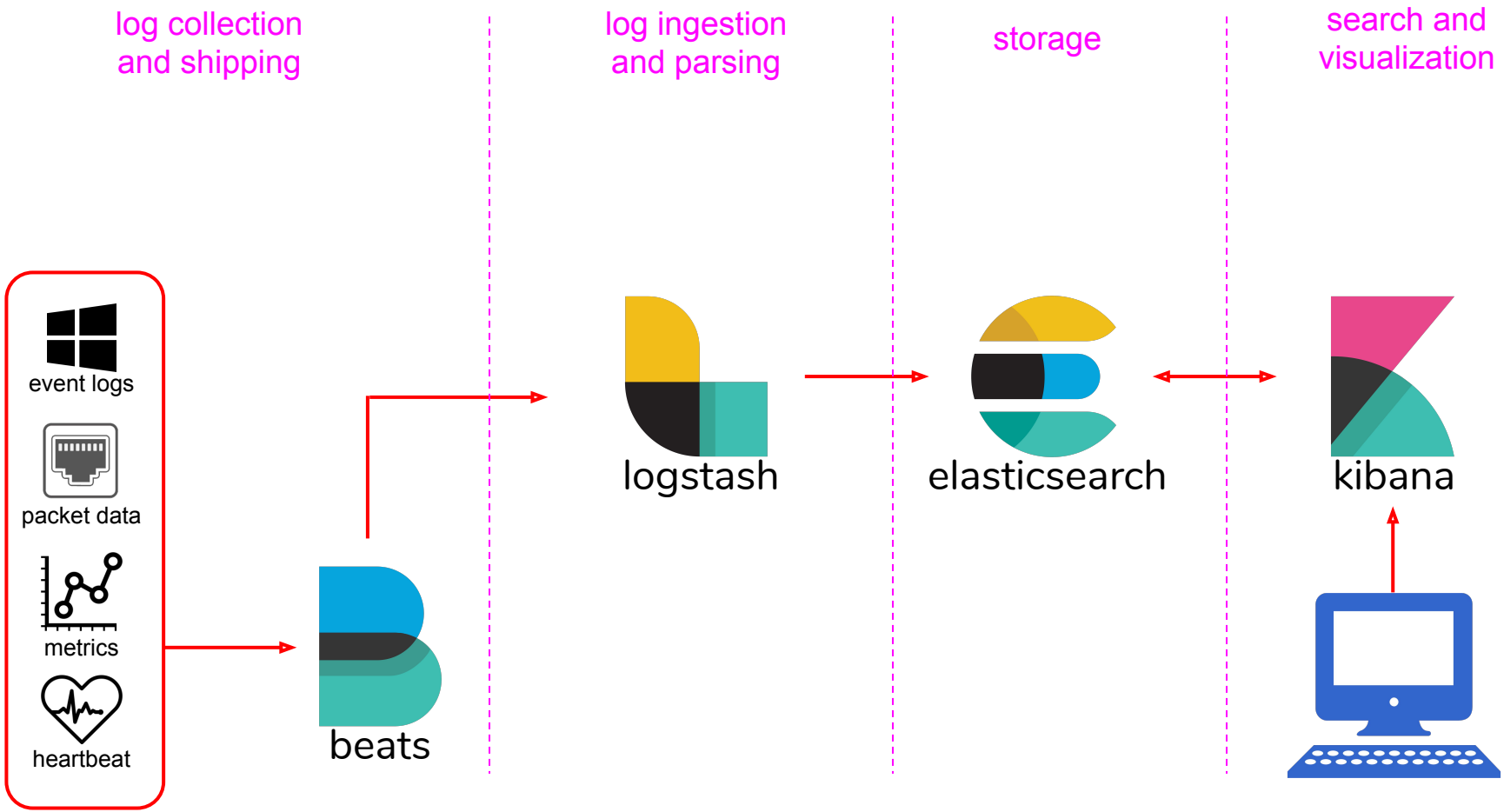
- Building a logging and alerting infrastructure
  - Intro to Docker and Elasticstack
  - Security Considerations
- Collecting and normalizing logs
  - Introduction to Beats (log collectors)
  - Collecting and logging from a variety of sources
  - Osquery intro
- Enriching logs and building baselines and inventories
  - Enriching logs from various sources of data
  - Creating baselines to identify anomalous behavior
- Creating effective alerts
  - Create basic alerts with ElastAlert
  - Implement event-based scoring
  - Introduction to Sigma

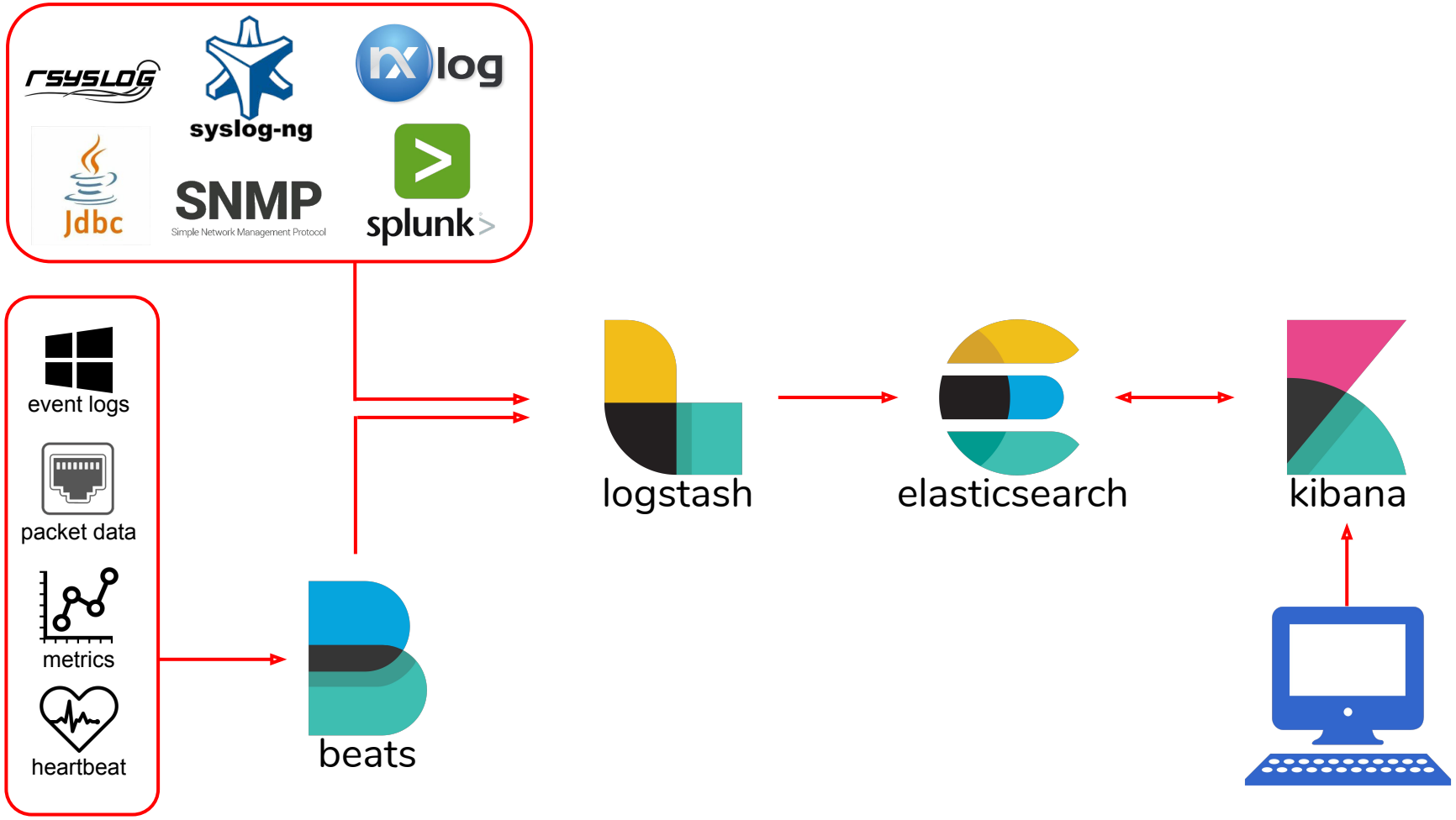

elasticsearch

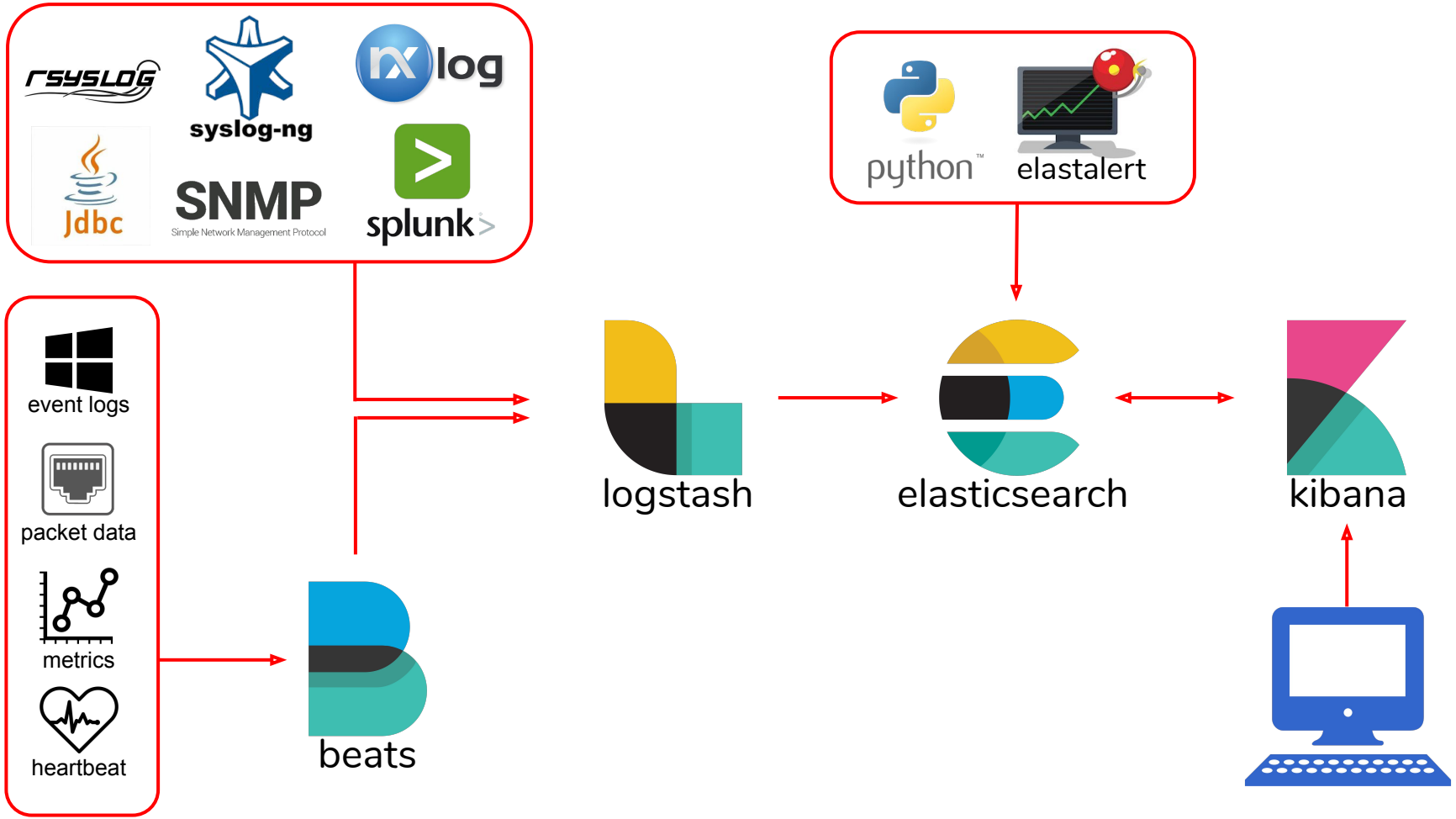

storage
search and visualization
elasticsearch
kibana


storage
search and visualization
elasticsearch
kibana


log ingestion
and parsing
storage
search and
visualization
logstash
elasticsearch
kibana


log collection
and shipping
log ingestion
and parsing
storage
search and
visualization
event logs
packet data
metrics
heartbeat
beats
logstash
elasticsearch
kibana


rsyslog
syslog-ng
nxlog
Jdbc
SNMP
Simple Network Management Protocol
splunk>
event logs
packet data
metrics
heartbeat
beats
logstash
elasticsearch
kibana


rsyslog
syslog-ng
nxlog
Jdbc
SNMP
Simple Network Management Protocol
splunk>
python
elastalert
event logs
packet data
metrics
heartbeat
beats
logstash
elasticsearch
kibana

Elasticsearch

- “For search”
- Built on Apache Lucene
- No schema
- Full-text search
- Highly available by design
- Built-in persistence
- REST API

Elasticsearch

Let’s get real for a minute...

Elasticsearch

First, the Bad News:

- Learning curve can be steep (but we'll help)
- Security is *NOT* baked-in (but it's getting there)
- Requires care and feeding
- Software is free, hardware and support are not

Elasticsearch

But wait, here's the Good News:

- Fast search (especially compared to other SIEMs)
- Open source and *mostly* free
- Quick to get started
- Can run on commodity hardware
- Tons of 3rd party support available

Elasticsearch

[vocabulary]

Cluster

- collection of one or more nodes (servers)
- holds the entirety of your data
- provides federated indexing and search capabilities
- identified by a unique name (default: “elasticsearch”)

Elasticsearch

[vocabulary]

Node

- single server that is part of your cluster
- stores your data
- participates in the cluster’s indexing and search
- has a unique name
- forms a cluster on its own if no other nodes are found

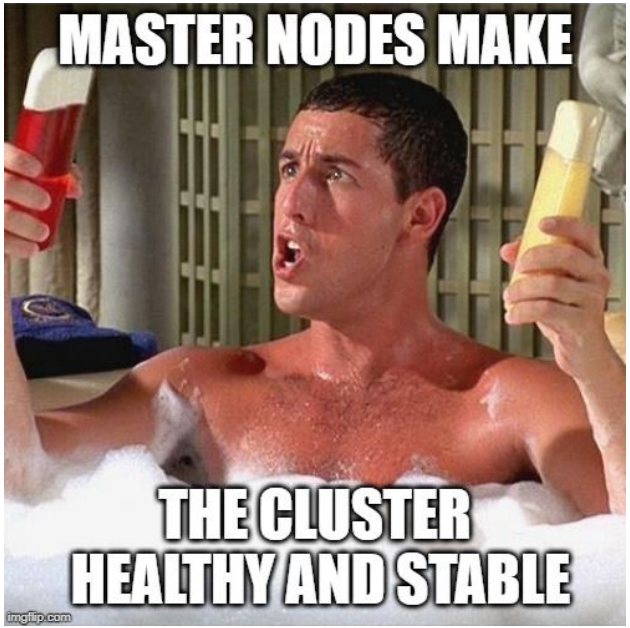
Elasticsearch

[vocabulary]

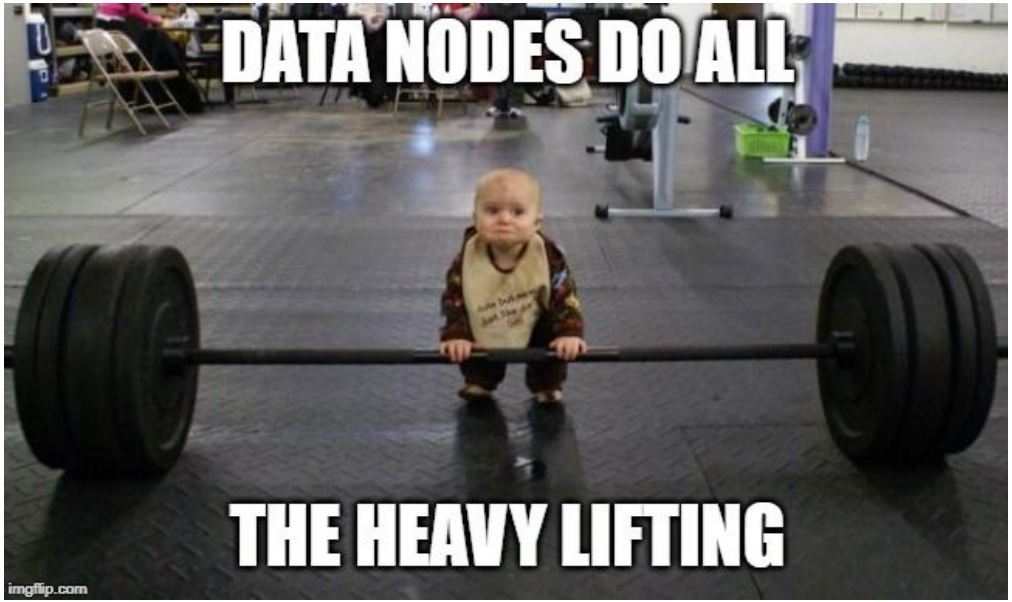
Node (cont'd)

- different 'roles' or purposes:
  - master
  - ingest
  - data
  - coordinating (kind of)

Elasticsearch



Elasticsearch



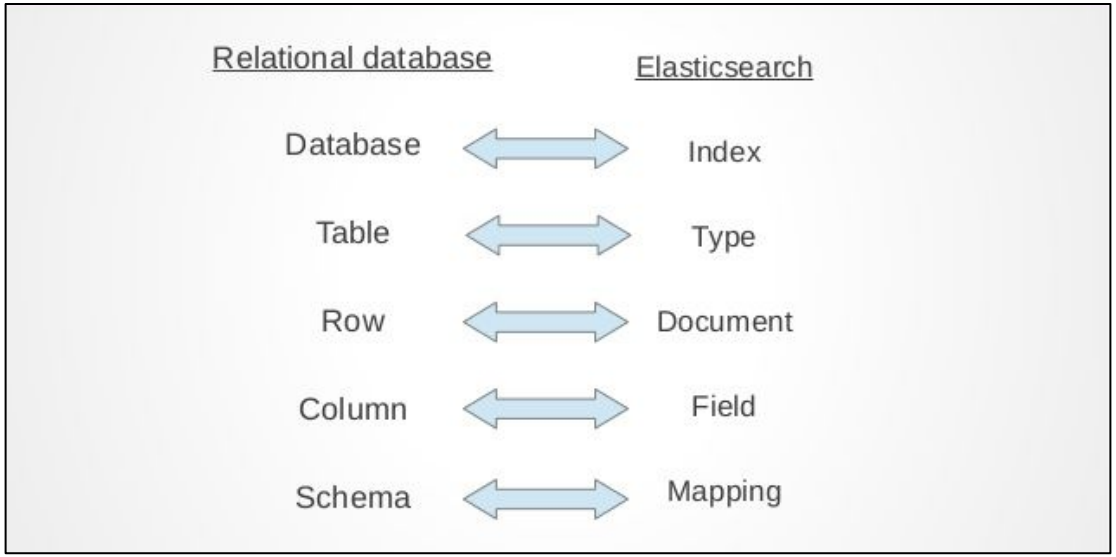
Elasticsearch

[We ran out of memes…]

Coordinating-only node pushes search requests out to all data nodes, then compiles the results before returning them. Every node is implicitly a coordinating node.

Ingest node allows for pre-processing, but we'll do that with logstash. It might be useful, depending on your use case.

Elasticsearch



https://www.devopsschool.com/blog/understanding-elasticsearch-keywords-and-terminology/

Elasticsearch

Removal of mapping types

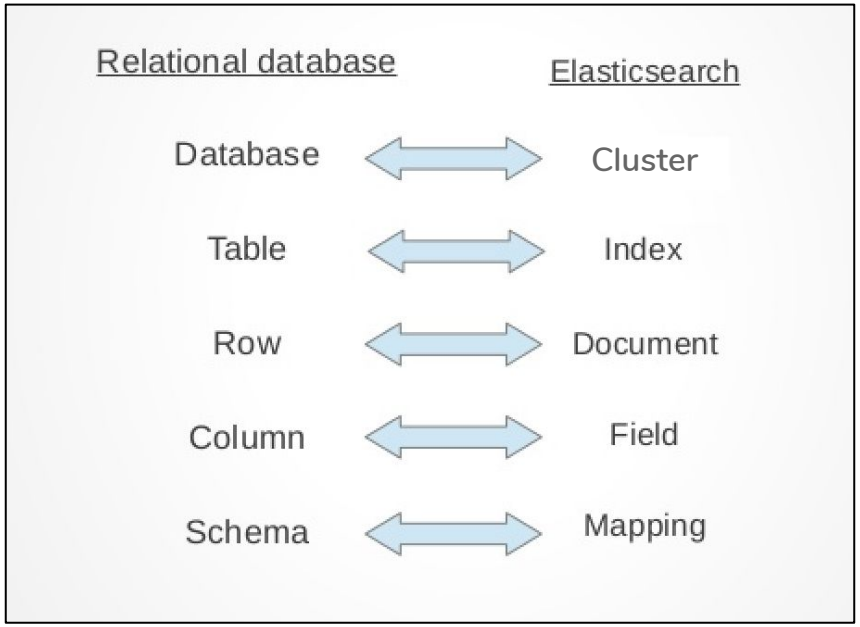
edit

IMPORTANT

Indices created in Elasticsearch 7.0.0 or later no longer accept a `_default_` mapping. Indices created in 6.x will continue to function as before in Elasticsearch 6.x. Types are deprecated in APIs in 7.0, with breaking changes to the index creation, put mapping, get mapping, put template, get template and get field mappings APIs.

https://www.elastic.co/guide/en/elasticsearch/reference/current/removal-of-types.html

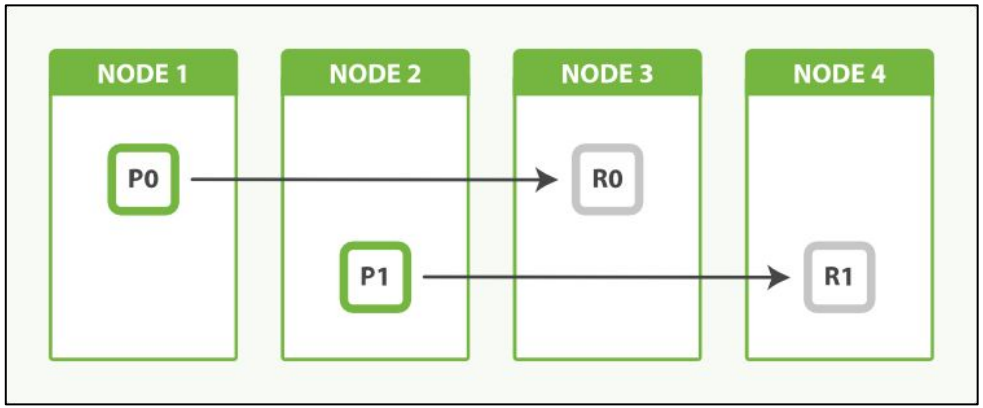
Elasticsearch



Elasticsearch

[vocabulary]

Shards: indexes are divided into sets of primary and replica shards, which are distributed across nodes.



https://www.elastic.co/guide/en/elasticsearch/guide/current/replica-shards.html

Elasticsearch

Security Considerations - TLS:

- Endpoints to Logstash
- Logstash to Elasticsearch nodes
- Elasticsearch to Elasticsearch
- Elasticsearch REST API
- Kibana to Elasticsearch
- Browser to Kibana

Elasticsearch

Security Considerations - Authentication:

- Logstash to Endpoints
- Logstash to Elasticsearch nodes
- Elasticsearch to Elasticsearch
- Elasticsearch REST API
- Kibana to Elasticsearch
- Browser to Kibana

Elasticsearch

Security Considerations - Authorization:

- Per cluster
- Per index
- Per document
- Per field
- CRUD actions

Elasticsearch

Security Considerations - Auditing:

- CRUD actions
- Logins

Elasticsearch

Security Considerations - Options:

- X-Pack: Free and paid versions from Elastic
- Search-Guard: Free and paid versions
- Open-Distro: Free/open source from Amazon
- Reverse Proxy: e.g. Nginx; offers TLS and authn

Elasticsearch

A few things to know:

- Plan to deploy three master nodes per cluster
- Keep shards under 50 GB
- Don’t allocate over 32 GB of memory to Java heap (because of compressed oops)
- Try to keep node storage around 4 TB (8 TB max)
- Calculate raw data size x 2.2 to approximate the size of indexed data on disk

Kibana

Search front-end to Elasticsearch

- Search using Lucene, DSL, or SQL-like query
- Visualize and create dashboards
- Monitor and manage your cluster
- Perform time series analysis

Kibana

Let’s take a tour…

[Open the Kibana shortcut on your Linux VM desktop, or navigate to http://<linux_vm_ip>:5601 from the browser of your choice on your VMWare host and follow along.]

Kibana

To explore a live system with plenty of sample data, visualizations, and dashboards visit:

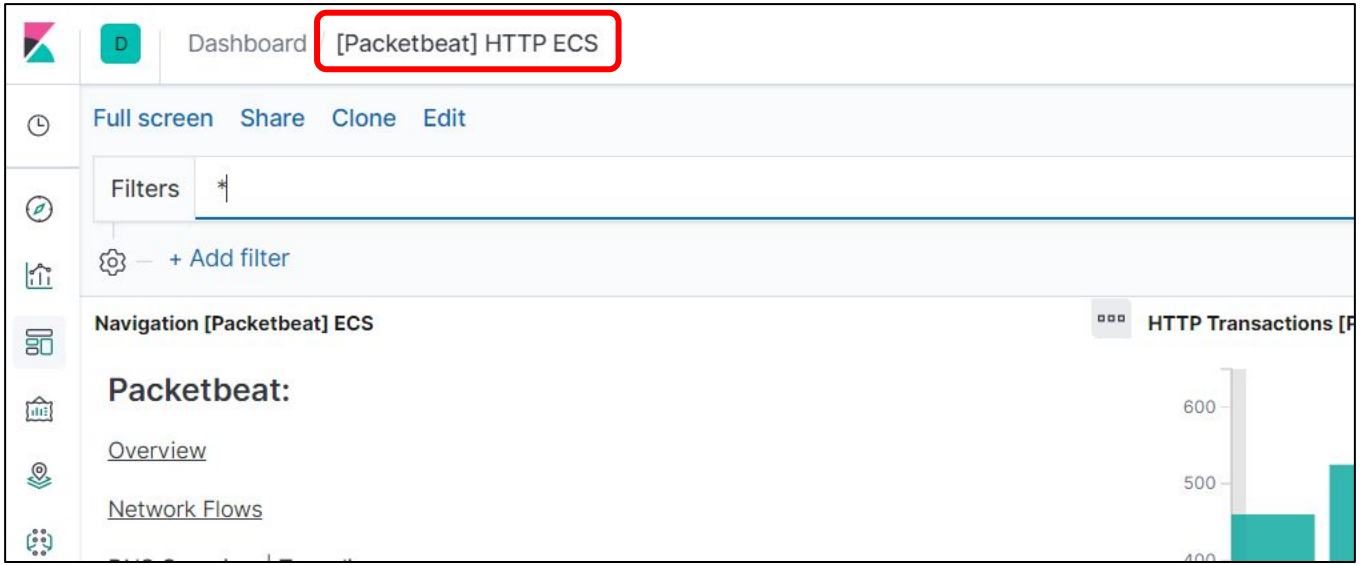
https://demo.elastic.co/app/kibana#/discover

Kibana

Want to use one of those fancy dashboards, but don’t want to build it from scratch?

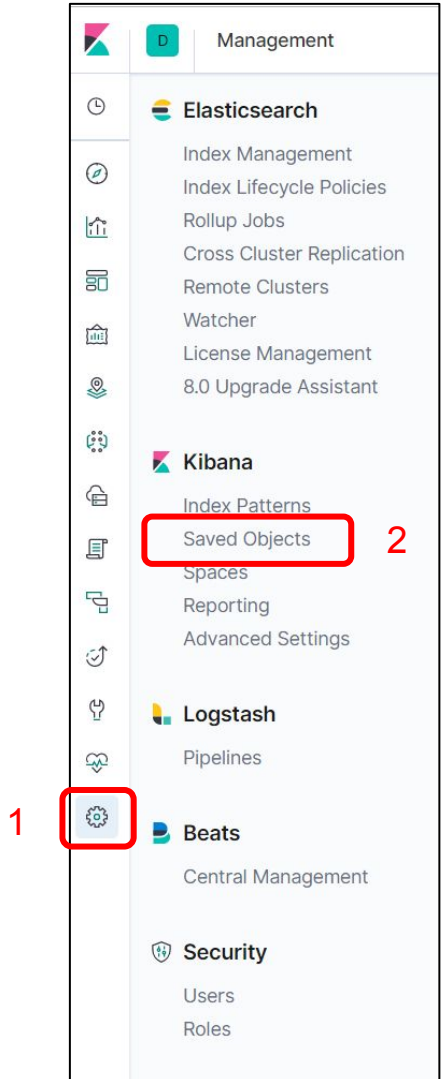
Kibana

Note the dashboard name



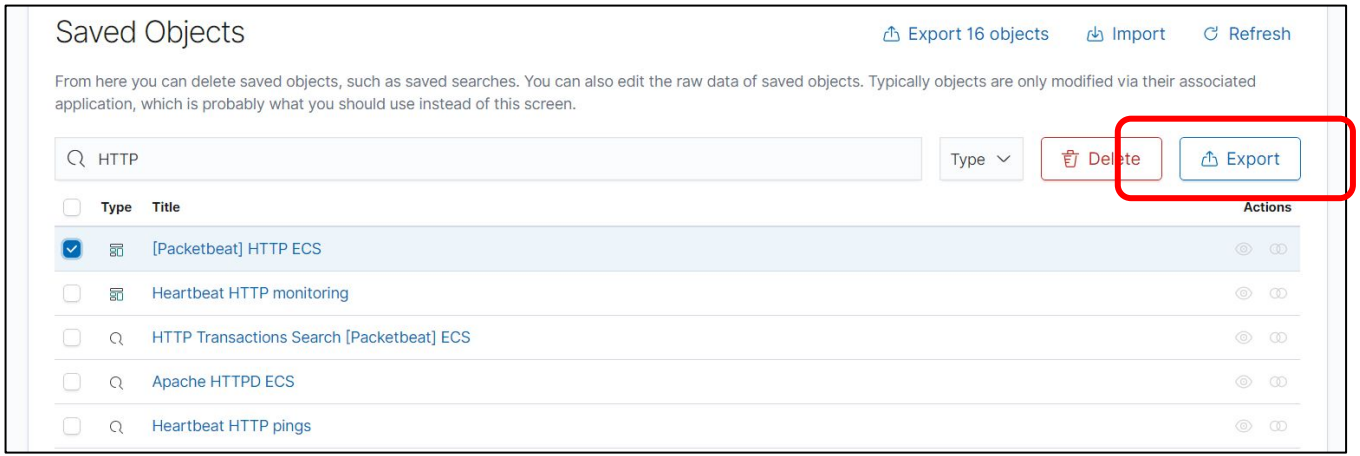
Kibana

1. Navigate to “Management”
2. Click “Saved Objects”



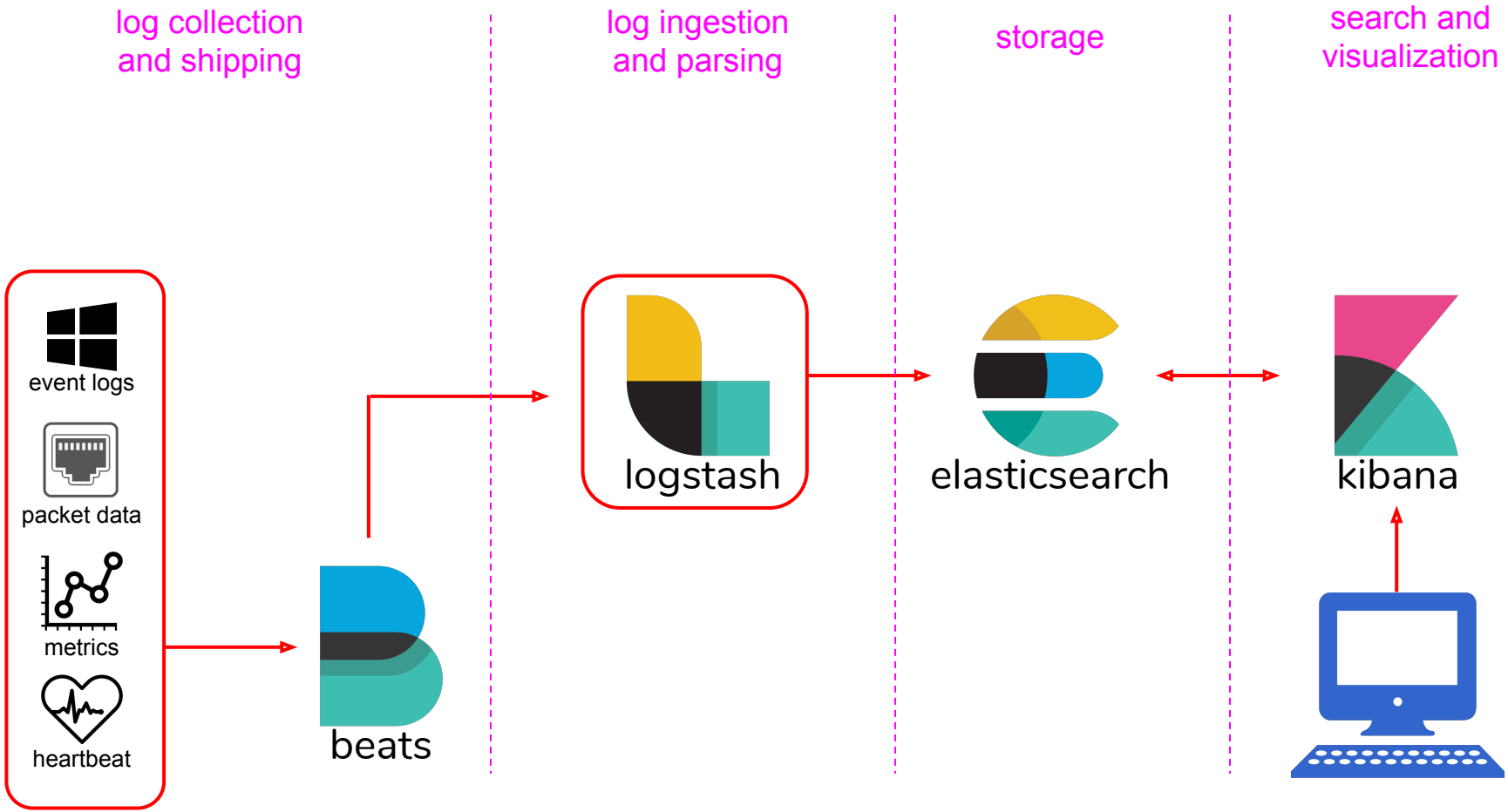
Kibana

Search for and select the dashboard name.
Then click “Export.”



Kibana

In your Kibana instance, go back to Management > Saved Objects and Import the json file you downloaded.

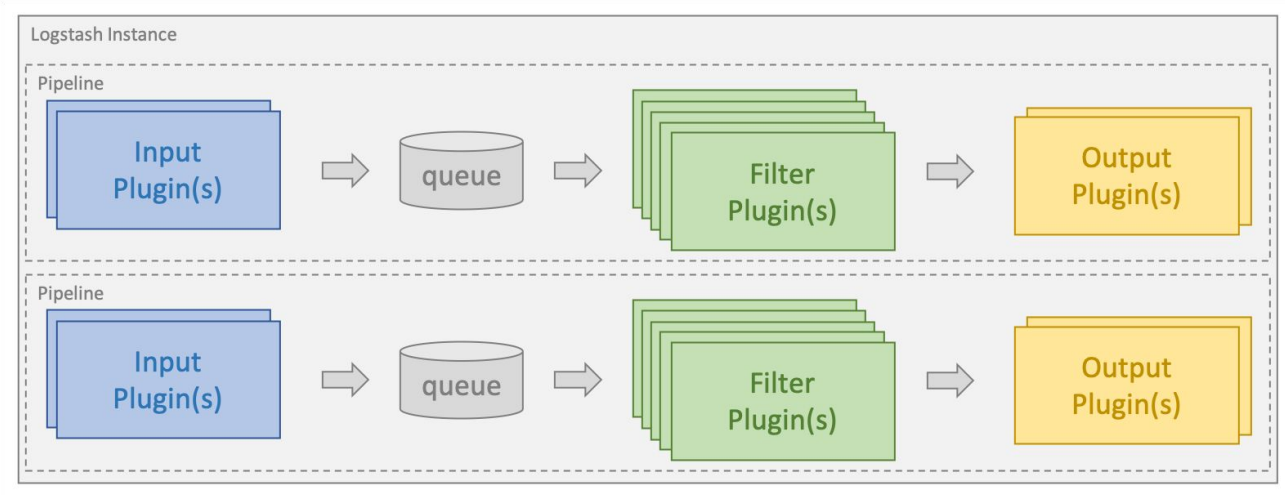

log collection
and shipping
log ingestion
and parsing
storage
search and
visualization
event logs
packet data
metrics
heartbeat
beats
logstash
elasticsearch
kibana

Logstash

- Collect, Parse, Transform Logs
- Horizontally scalable data processing
- Open source
- Input, filter, and output plugins
- Over 200 plugins
- Reliability and resiliency, disk based queue

Logstash

- **Input:** Listen and accept logs
- **Filter:** Filter, normalize, and enrich logs
- **Output:** Send logs to external application or system



https://www.elastic.co/blog/a-practical-introduction-to-logstash

Logstash

Input plugins:

- Beats - winlogbeat, filebeat, packetbeat, etc
- TCP/UDP - Listen on ports for data
- File - Read data from files on disk
- Database - elasticsearch, jdbc, sqlite
- Message Brokers - kafka, redis, rabbitmq
- Others - netflow, snmp, http (REST APIs)

Logstash

Filter plugins:

- Parsing - regex (grok), json, csv, kv, syslog
- Transforms - convert, rename, remove, date, drop
- Enrichment - geoip, dns, rest, lookups
- Ruby - Execute ruby code

Logstash

Output plugins:

- Storage - elasticsearch, mongodb, s3
- Network - tcp, udp, http
- Stdout - for troubleshooting
- Message broker - kafka, rabbitmq, redis

Logstash

Pipelines:

- Log processing organized into pipelines
- One or more config files in each
- Automatic reloads
- Define queues, threads

Logstash

Pipeline config example:

```
# cat /usr/share/logstash/config/pipelines.yml

- pipeline.id: winlogbeat
  path.config: "/usr/share/logstash/pipeline/winlogbeat"
- pipeline.id: filebeat
  path.config: "/usr/share/logstash/pipeline/filebeat"
- pipeline.id: packetbeat
  path.config: "/usr/share/logstash/pipeline/packetbeat"
```

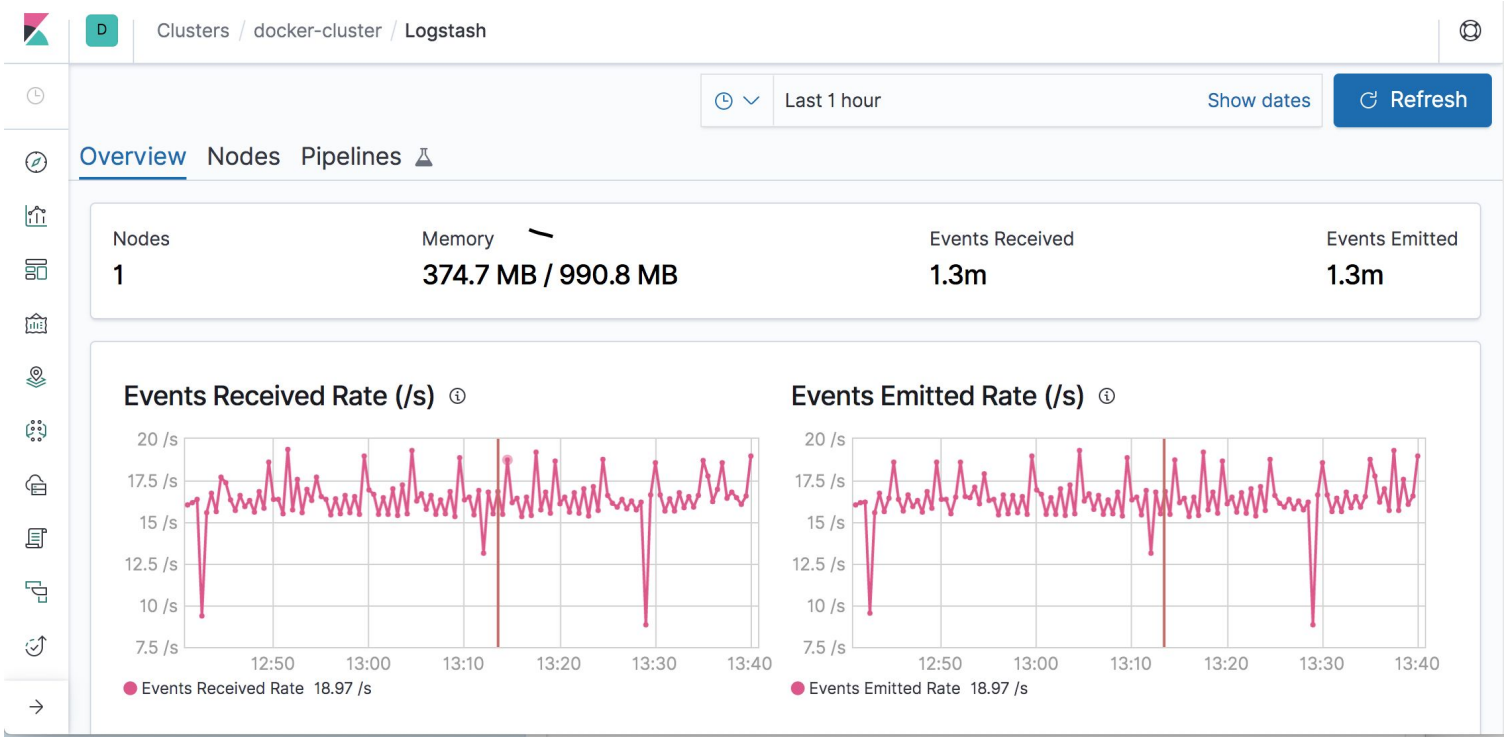
Logstash

Config example:

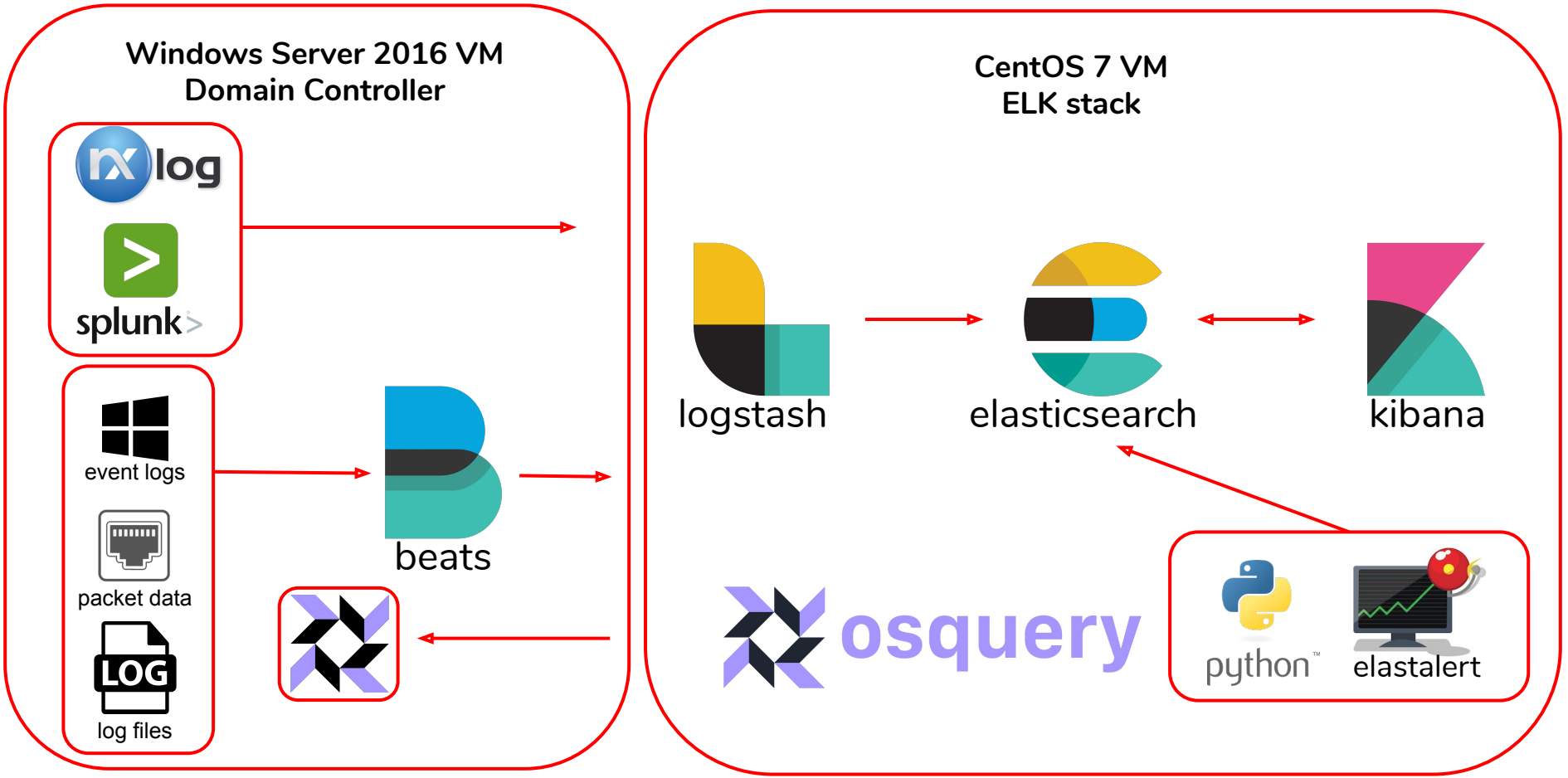
```
# cat /usr/share/logstash/pipeline/winlogbeat/logstash.conf

input {
  beats {
    port => 5044
  }
}
output {
  elasticsearch {
    hosts => ["es01:9200"]
    index => "logstash-%{[agent][type]}-%{+YYYY.MM.dd}"
  }
}
```

Logstash

Monitoring:



Lab Environment



Windows VM details

Config:	**Software:**
• Windows Server 2016 • Domain Controller and DNS • GPO - Advanced Audit Policy • PowerShell logging • Sysmon (SwiftOnSecurity) • Autoruns to event log • DNS debug logging	• Chocolatey • Winlogbeat • Packetbeat (winpcap) • Filebeat • osquery • nxlog • Splunk forwarder • Sysinternals • Notepad++

Scripts, configs, GPO source: https://github.com/clong/DetectionLab

Windows VM details

Group Policy Advanced Auditing Policy:

- View current settings
  - auditpol.exe /get /category:*
  - gpresult /v
- Lab settings:
  - gpmc.msc > Domain Controllers Enhanced Auditing Policy
  - c:\temp\scripts\GPO\Domain_Controllers_Enhanced_Auditing_Policy_2
- Recommendations:
  - Sean Metcalf's recommendations: https://adsecurity.org/?p=3377
  - Microsoft: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/audit-policy-recommendations

Scripts and GPO source: https://github.com/clong/DetectionLab

Windows VM details

Sysmon:

- Part of Microsoft's Sysinternals suite - https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon
- Writes to the Windows event log
- Provides EDR style logging
- Configs
  - SwiftOnSecurity - https://github.com/SwiftOnSecurity/sysmon-config
  - Olaf Hartong - https://github.com/olafhartong/sysmon-modular
- Lab settings:
  - Install script: C:\temp\scripts\install-sysinternals.ps1

Install script source: https://github.com/clong/DetectionLab

Windows VM details

Beats and agents:

- Winlogbeat
  - Config: c:\ProgramData\chocolatey\lib\winlogbeat\tools\winlogbeat.yml
- Packetbeat
  - Config: c:\ProgramData\chocolatey\lib\packetbeat\tools\packetbeat.yml
- Filebeat
  - Config: c:\ProgramData\chocolatey\lib\filebeat\tools\filebeat.yml
- nxlog
  - Config: C:\Program Files (x86)\nxlog\conf\nxlog.conf
- Splunk forwarder
  - C:\Program Files\SplunkUniversalForwarder\etc\system\local\

Windows VM details

Beats overview

- Winlogbeat
  - Stream Windows event logs to Elasticsearch or Logstash
  - Data in a structured format to make filtering and aggregating easy
- Packetbeat
  - Lightweight packet analyzer that sends data to Logstash or Elasticsearch
  - Supports traffic flow, HTTP, DNS, MYSQL, and more
- Filebeat
  - Lightweight way to forward and centrailize logs
  - Internal modules (auditd, Apache, NGINX, System, MySQL, and more)

Windows VM details

Other agents

- NXLog
  - Supports input from Windows EventLog, files, databases, tcp/udp, exec and stdout
  - Can output json, csv, gelf and other structured formats
- Splunk
  - Supports multiple outputs, send data to Splunk and other data to ELK
  - Many types of inputs and free apps
- osquery
  - Real-time insight into the current state of your infrastructure
  - Works on Windows, Mac OS X, Ubuntu, Cent OS, and more
  - Ad-hoc or recurring, scheduled queries

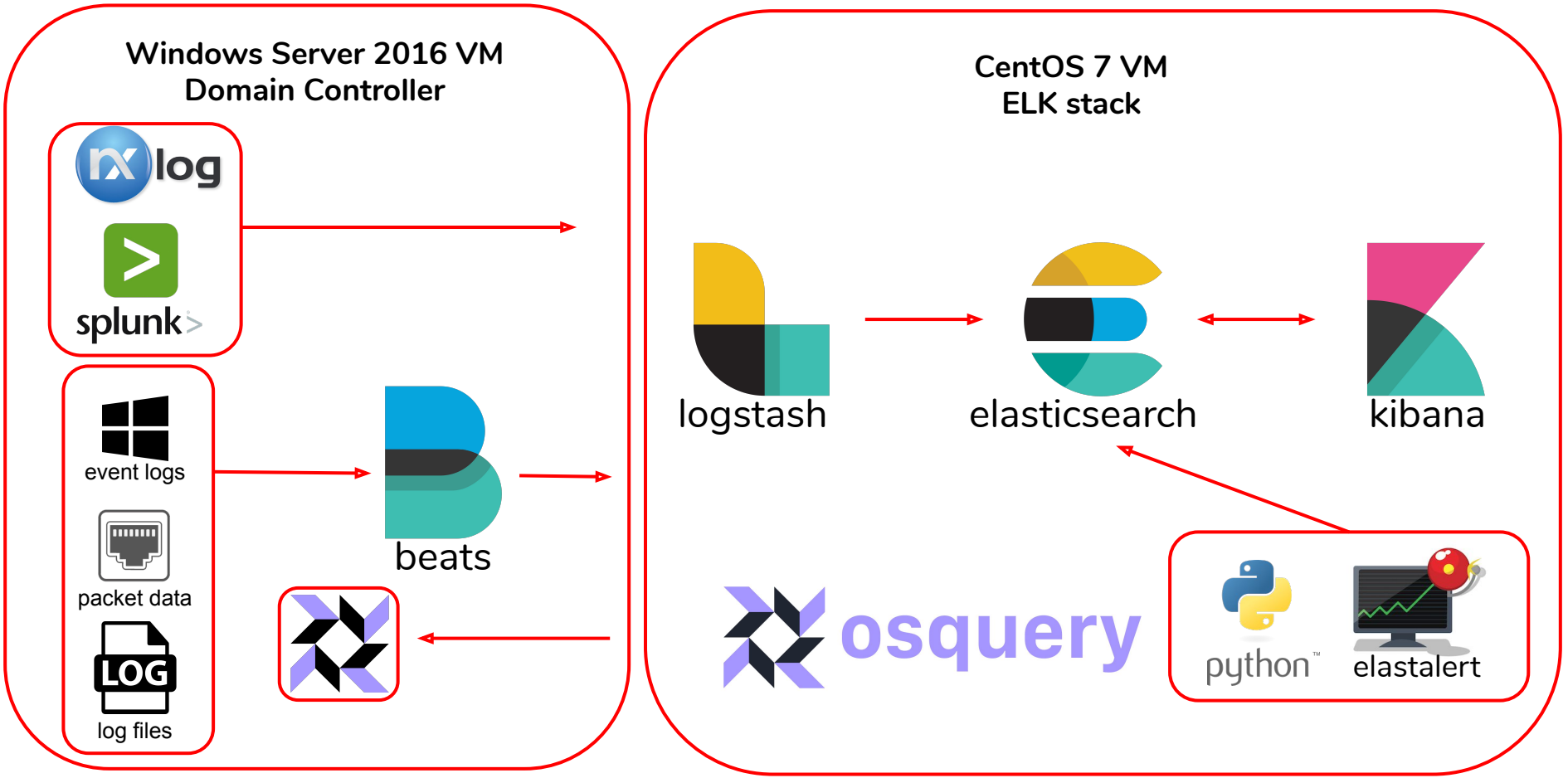
Windows VM details

Other configuration:

- PowerShell logging GPO
  - gpmc.msc > PowerShell logging
  - Logs: c:\pslogs\
- Command line logging
  - Windows Settings\Admin Templates\System\Audit Process Creation
  - Security EventID: 4688
- DNS Debug logging
  - dnsmgmt.msc > Right click DC > Properties > Debug Logging tab
  - c:\temp\scripts\configure-DNSdebuglogging.ps1
  - Logs: c:\dnslogs\
- Autoruns to Eventlog
  - https://github.com/palantir/windows-event-forwarding/tree/master/AutorunsToWinEventLog

GPO source: https://github.com/clong/DetectionLab

Lab Environment



Logstash

Parsing: Grok

- Parse unstructured log data
- Built on regular expressions
- Useful for syslog, webservers, and any Cisco product
- Logstash ships with about 120 patterns by default
- Ability to create your own patterns
- Example: %{NUMBER:duration} %{IP:client}

Grok patterns: https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns

Logstash

Parsing: Grok

```
# cat /var/log/http.log
55.3.244.1 GET /index.html 15824 0.043

# cat /usr/share/logstash/pipeline/httpd/logstash.conf
input {
  file {
    path => "/var/log/http.log"
  }
}
filter {
  grok {
    match => { "message" => "%{IP:client} %{WORD:method} %{URIPATHPARAM:request}
%{NUMBER:bytes} %{NUMBER:duration}" }
  }
}
```

Source: https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html

Logstash

Parsing:

- Structured formats: CSV, XML, key value, JSON
- Logstash plugins easily parse these formats

```
filter {
  json {
    source => message
  }
}
```

```
filter {
 csv {
    columns = [“date”, “user”, “src”, “dest”, “message”]
  }
}
```

Logstash

Parsing: Adding Tags

```
filter {
 if [source_ip] {
   if [source_ip] =~ "^10\." or [source_ip] =~ "192\.168\." or [source_ip] =~ "172\.(1[6-9]|2[0-9]|3[0-1])\." {
     mutate {
       add_tag => [ "internal_source" ]
     }
   } else {
     mutate {
       add_tag => [ "external_source" ]
     }
   }
 if "internal_source" in [tags] and "internal_destination" in [tags] {
     mutate { add_tag => [ "internal_only" ] }
   }
}
```

Source: https://github.com/HASecuritySolutions/Logstash/blob/master/configfiles/8200_postprocess_tagging.conf

“Query your endpoints like a SQL database”

- Developed at Facebook
- Fast, free, and runs everywhere
- Open source
- Continuously tested for memory leaks, thread safety, etc

Kolide Fleet

Open Source osquery Manager

- Open source and free, though there's a paid cloud version
- Provides a centralized GUI front-end to osquery on endpoints

Kolide Fleet

Let's take a tour...

[Open the Kolide shortcut on your Linux VM desktop, or navigate to http://<linux_vm_ip>:8443 from the browser of your choice on your VMWare host and follow along.]

username: **GuerillaBT**
password: **bsid3s!**

Logstash

Normalizing log data:

- Why?
  - Provide a consistent and customizable way to structure your data in Elasticsearch
  - s_ip, source-ip, client.ip, src
- How?
  - Elastic Common Schema - https://www.elastic.co/blog/introducing-the-elastic-common-schema
  - Splunk CIM - https://docs.splunk.com/Documentation/CIM/latest/User/Overview
  - Cyb3rWard0g - https://github.com/Cyb3rWard0g/OSSEM

Logstash

Enrichment:

- DNS - Forward and reverse lookups with caching
- GeoIP - Built on Maxmind GeoLite2 database
- Elasticsearch - Add previously logged data into current
- Translate - Lookups based on yaml, json, csv
- JDBC - Add data from remote databases
- User Agent - Parse useragent for OS, application,etc
- Others - memcache, http

Logstash

Enrichment: Examples

```
filter {
  dns {
    reverse => [ "source_host" ]
    action => "replace"
  }
}
```

```
filter {
  geoip {
    source => "clientip"
  }
}
```

Source: https://www.elastic.co/guide/en/logstash/current/lookup-enrichment.html

Logstash

Enrichment: Examples

Malware domain list:
$ head malware.yaml
"213.155.12.XXX/sec/bin/upload/v1crypted.exe": "true"
"128.134.30.XXX/w.exe" : "true"
"114.203.87.XXX/help.asp" : "true"
	filter {
	  translate {
	  field => "url"
	  destination => "malware"
	  dictionary_path => "malware.yaml"
	  }
	}

Input:
{ "url" : "128.134.30.XXX/w.exe" }
	Output:
	{
	  "@timestamp" => 2018-01-15T09:53:10.829Z,
	  "malware" => "true",
	  "url" => "128.134.30.XXX/w.exe",
	  "@version" => "1",
	  "host" => "localhost"
	}

Source: https://www.elastic.co/blog/elasticsearch-data-enrichment-with-logstash-a-few-security-examples

Logstash

Enrichment: Examples

List Domain Admins:

```
PS C:\> $Group = “Administrators”
Get-ADGroupMember -Recursive $Group `
| select samaccountname, @{N="Group";e={"$Group"}} `
| %{Convertto-json -compress $_} | out-file .\groups.json
```

Output:

```
{"samaccountname":"admin","Group":"Administrators"}
{"samaccountname":"Administrator","Group":"Administrators"}
```

```
filter {
  translate {
    field => "user"
    destination => "user_type"
    dictionary_path => "groups.json"
  }
}
```

Logstash

Enrichment: Domain Stats

- Web API to deliver domain information from whois and Alexa/Cisco Top 1 million ranking
- Ability to cache and preload data
- https://github.com/MarkBaggett/domain_stats

Logstash

Enrichment: Elasticsearch filter

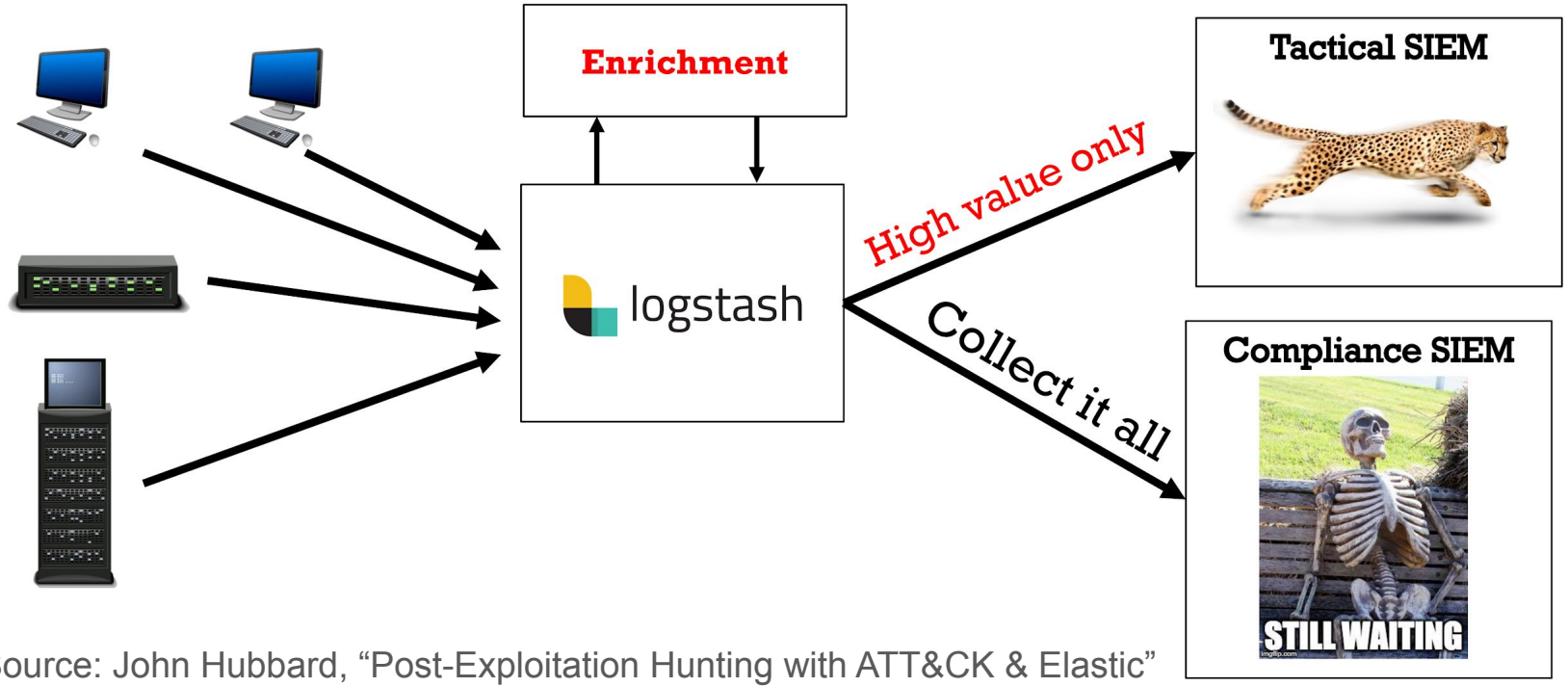
- Search Elasticsearch for a previous log event and fields to current event
- Ability to join and enrich multiple data sets
- Might slow down high volume sources

Logstash

Enrichment: Examples

- Vulnerability Scanner output in ELK
  - https://github.com/HASecuritySolutions/VulnWhisperer
- LOLBAS - https://github.com/LOLBAS-Project/LOLBAS
- DeepBlueCLI - https://github.com/sans-blue-team/DeepBlueCLI
- Free threat intelligence feeds
  - https://github.com/hslatman/awesome-threat-intelligence

Logstash



- Source: John Hubbard, “Post-Exploitation Hunting with ATT&CK & Elastic” https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1511995081.pdf

Logstash

Adding command line length lab

- http://bit.ly/bsides-cmd-lab

Elastalert

- Developed and maintained by Yelp
- Designed to be reliable, modular and simple
- Alerts on anomalies, spike and patterns of interest
- Free (Apache License) and open source
- Python ftw

Elastalert Global Configuration [config.yml]

```
rules_folder: example_rules
run_every:
  minutes: 1
buffer_time:
  minutes: 15
es_host: elasticsearch.example.com
es_port: 9200
aws_region: us-east-1
profile: test
es_url_prefix: elasticsearch
use_ssl: True
verify_certs: True
es_send_get_body_as: GET
es_username: someusername
es_password: somepassword
verify_certs: True
ca_certs: /path/to/cacert.pem
client_cert: /path/to/client_cert.pem
client_key: /path/to/client_key.key
writeback_index: elastalert_status
alert_time_limit:
  days: 2
```

Elastalert Global Configuration [config.yml]

Elastalert Global Config Example	Description
rules_folder: example_rules	This is the folder that contains the rule yaml files. Any .yaml file will be loaded as a rule
run_every: minutes: 1	How often ElastAlert will query Elasticsearch. The unit can be anything from weeks to seconds
buffer_time: minutes: 15	ElastAlert will buffer results from the most recent period of time, in case some log sources are not in real time
es_host: elasticsearch.com es_port: 9200	The Elasticsearch hostname for metadata writeback. Note that every rule can have its own Elasticsearch host. Rules can overwrite any setting in config.yaml including ES properties.
aws_region: us-east-1	[optional] The AWS region to use. Set this when using AWS-managed elasticsearch
profile: test	[optional] The AWS profile to use. Use this if you are using an aws-cli profile.
es_url_prefix: elasticsearch	[optional] URL prefix for Elasticsearch
use_ssl: True	[optional] Connect with TLS to Elasticsearch
verify_certs: True	[optional] Verify TLS certificates
es_send_get_body_as: GET	[optional] GET request with body is the default option for Elasticsearch. If it fails for some reason, you can pass 'GET', 'POST' or 'source'.
es_username: someusername es_password: somepassword	[optional] Option basic-auth username and password for Elasticsearch
ca_certs: /path/to/cacert.pem client_cert: /path/to/client_cert.pem client_key: /path/to/client_key.key	[optional] A pem file containing both cert and key for client verify_certs: True
writeback_index: elastalert_status	The index on es_host which is used for metadata storage. This can be a unmapped index, but it is recommended that you run elastalert-create-index to set a mapping
alert_time_limit: days: 2	If an alert fails for some reason, ElastAlert will retry sending the alert until this time period has elapsed

https://posts.specterops.io/what-the-helk-sigma-integration-via-elastalert-6edf1715b02

Elastalert Alert Types

Alert Type	Description
Command	The command alert allows you to execute an arbitrary command and pass arguments or stdin from the match. Arguments to the command can use Python format string syntax to access parts of the match.
Email	This alert will send an email. It connects to an smtp server located at smtp_host, or localhost by default.
JIRA	The JIRA alerter will open a ticket on jira whenever an alert is triggered. You must have a service account for ElastAlert to connect with.
OpsGenie	OpsGenie alerter will create an alert which can be used to notify Operations people of issues or log information. An OpsGenie API integration must be created in order to acquire the necessary opsgenie_key rule variable.
SNS	The SNS alerter will send an SNS notification. The body of the notification is formatted the same as with other alerters. The SNS alerter uses boto3 and can use credentials in the rule yaml, in a standard AWS credential and config files, or via environment variables.
HipChat	HipChat alerter will send a notification to a predefined HipChat room. The body of the notification is formatted the same as with other alerters.
Slack	Slack alerter will send a notification to a predefined Slack channel. The body of the notification is formatted the same as with other alerters.
Telegram	Telegram alerter will send a notification to a predefined Telegram username or channel. The body of the notification is formatted the same as with other alerters.
GoogleChat	GoogleChat alerter will send a notification to a predefined GoogleChat channel. The body of the notification is formatted the same as with other alerters.
Debug	The debug alerter will log the alert information using the Python logger at the info level. It is logged into a Python Logger object with the name elastalert that can be easily accessed using the getLogger command.
Stomp	This alert type will use the STOMP protocol in order to push a message to a broker like ActiveMQ or RabbitMQ. The message body is a JSON string containing the alert details. The default values will work with a pristine ActiveMQ installation.
theHive	theHive alert type will send JSON request to theHive (Security Incident Response Platform) with TheHive4py API

https://posts.specterops.io/what-the-helk-sigma-integration-via-elastalert-6edf1715b02

Elastalert Alert Types

Several rule types are included with ElastAlert:

- **Frequency**: Match where there are X events in Y time
- **Spike**: Match when the rate of events increases or decreases
- **Flatline**: Match when there are less than X events in Y time
- **Blacklist/Whitelist**: Match when a certain field matches a blacklist/whitelist
- **Any**: “Match on any event matching a given filter
- **Change**: Match when a field has two different values within some time

Elastalert Example Alert

Example Frequency Rule

(Required fields)

- **es_host**: Elasticsearch hostname
- **es_port**: Elasticsearch HTTP port
- **name**: arbitrary name you choose
- **type**: Elastalert rule type
- **index:** index to search (wildcard supported)
- **num_events:** number of events to match within the timeframe below
- **timeframe**: timeframe within which events must occur to trigger the alert
- **filter:** Elasticsearch filters to find events
- **alert:** alert type to use if alert is triggered
- **email**: required field for “email” alert type

```
1  es_host: es01
2  es_port: 9200
3
4  name: Example Rule
5  type: frequency
6
7  index: logstash-*
8
9  num_events: 50
10 timeframe:
11   hours: 4
12
13 filter:
14 - term:
15     some_field: "some_value"
16
17 alert:
18 - "email"
19
20 email:
21 - "elastalert@example.com"
22
```

For more about filter syntax (using DSL): http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/query-dsl.html

- Generic Signature Format for SIEM Systems
- Developed by Florian Roth and Thomas Patzke
- Rule format is very flexible, easy to write and applicable to any type of log file
- ~200 rules available in the project
- SIEM searches in Sigma to avoid a vendor lock-in
- Free and open source
- https://github.com/Neo23x0/sigma

SIGMA Converter (Sigmac)

Output Target	Description
arcsight	Converts Sigma rule into ArcSight saved search.
es-qs	Converts Sigma rule into Elasticsearch query string. Only searches, no aggregations
es-dsl	ElasticSearch DSL backend
kibana	Converts Sigma rule into Kibana JSON Configuration files (searches only).
xpack-watcher	Converts Sigma Rule into X-Pack Watcher JSON for alerting
elastalert	Elastalert backend
graylog	Converts Sigma rule into Graylog query string. Only searches, no aggregations.
logpoint	Converts Sigma rule into LogPoint query
grep	Generates Perl compatible regular expressions and puts 'grep -P' around it
netwitness	Converts Sigma rule into NetWitness saved search. Contributed by @tuckner
powershell	Converts Sigma rule into PowerShell event log cmdlets.
qradar	Converts Sigma rule into Qradar saved search.
qualys	Converts Sigma rule into Qualys saved search.
splunk	Converts Sigma rule into Splunk Search Processing Language (SPL).
splunkxml	Converts Sigma rule into XML used for Splunk Dashboard Panels
sumologic	Converts Sigma rule into Sumologic rule format
fieldlist	List all field names from given Sigma rules for creation of a field mapping configuration.
wdatp	Converts Sigma rule into Windows Defender ATP Hunting Queries.

https://posts.specterops.io/what-the-helk-sigma-integration-via-elastalert-6edf1715b02

SIGMA Field Mapping

```
logsources:
    windows-sysmon:
        product: windows
        service: sysmon
        index: logs-endpoint-winevent-sysmon-*
defaultindex: logs-*
fieldmappings:
    EventID: event_id
    ParentImage: process_parent_path
    CommandLine: process_command_line
    TargetObject: registry_key_path
    EventType: event_type
```

sysmon_stickykey_like_backdoor.yml

```
1  alert:
2  - debug
3  description: Detects the usage and installation of a backdoor that uses an option
4    to register a malicious debugger for built-in tools that are accessible in the login
5    screen
6  filter:
7  - query:
8      query_string:
9        query: (event_id:"1" AND process_parent_path:("*\\winlogon.exe") AND process_command_line:("*\\cmd.exe
10         sethc.exe *" "*\\cmd.exe utilman.exe *" "*\\cmd.exe osk.exe *" "*\\cmd.exe
11         Magnify.exe *" "*\\cmd.exe Narrator.exe *" "*\\cmd.exe DisplaySwitch.exe *"))
12         OR (event_id:"13" AND registry_key_path:("*\\SOFTWARE\\Microsoft\\Windows
13         NT\\CurrentVersion\\Image File Execution Options\\sethc.exe\\Debugger" "*\\SOFTWARE\\Microsoft\\Windows
14         NT\\CurrentVersion\\Image File Execution Options\\utilman.exe\\Debugger" "*\\SOFTWARE\\Microsoft\\Windows
15         NT\\CurrentVersion\\Image File Execution Options\\osk.exe\\Debugger" "*\\SOFTWARE\\Microsoft\\Windows
16         NT\\CurrentVersion\\Image File Execution Options\\Magnify.exe\\Debugger" "*\\SOFTWARE\\Microsoft\\Windows
17         NT\\CurrentVersion\\Image File Execution Options\\Narrator.exe\\Debugger"
18         "*\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution
19         Options\\DisplaySwitch.exe\\Debugger") AND event_type:"SetValue"))
20 index: logs-endpoint-winevent-sysmon-*
21 name: Sticky-Key-Like-Backdoor-Usage_0
22 priority: 1
23 realert:
24   minutes: 0
25 type: any
26
```

https://posts.specterops.io/what-the-helk-sigma-integration-via-elastalert-6edf1715b02

- Datasets
  - https://github.com/Cyb3rWard0g/mordor
  - https://github.com/sbousseaden/EVTX-ATTACK-SAMPLES
  - https://github.com/sans-blue-team/DeepBlueCLI/tree/master/evtx

- Other SIEM/ELK projects
  - Detection Lab - https://github.com/clong/DetectionLab
  - HELK - https://github.com/Cyb3rWard0g/HELK
  - SOF-ELK - https://github.com/philhagen/sof-elk
  - SANS555 - https://github.com/HASecuritySolutions
  - SecurityOnion - https://github.com/Security-Onion-Solutions/security-onion
  - RockNSM - https://rocknsm.io/