Module

Shipping Log Data

Topics

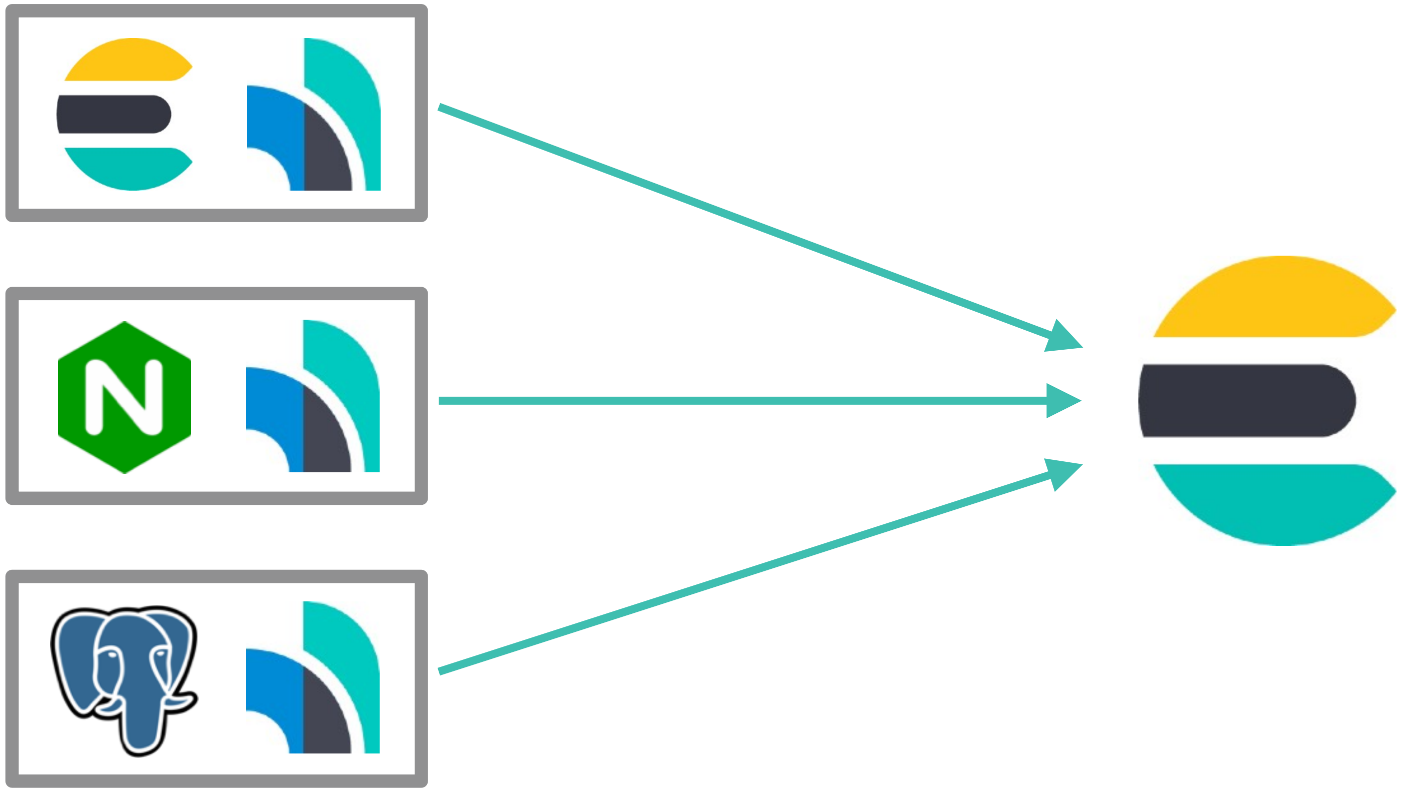
- Filebeat Architecture
- Modules
- Resilience
- Multiline Processing

Lesson 1

Filebeat Architecture

Filebeat

- Installed as an agent on all servers from which you want to gather logs
- Monitors log directories or specific log files
- Tails these input log files and ships log entries to an output



Can also ship log entries to Logstash, Kafka, Redis and more.

Why Filebeat?

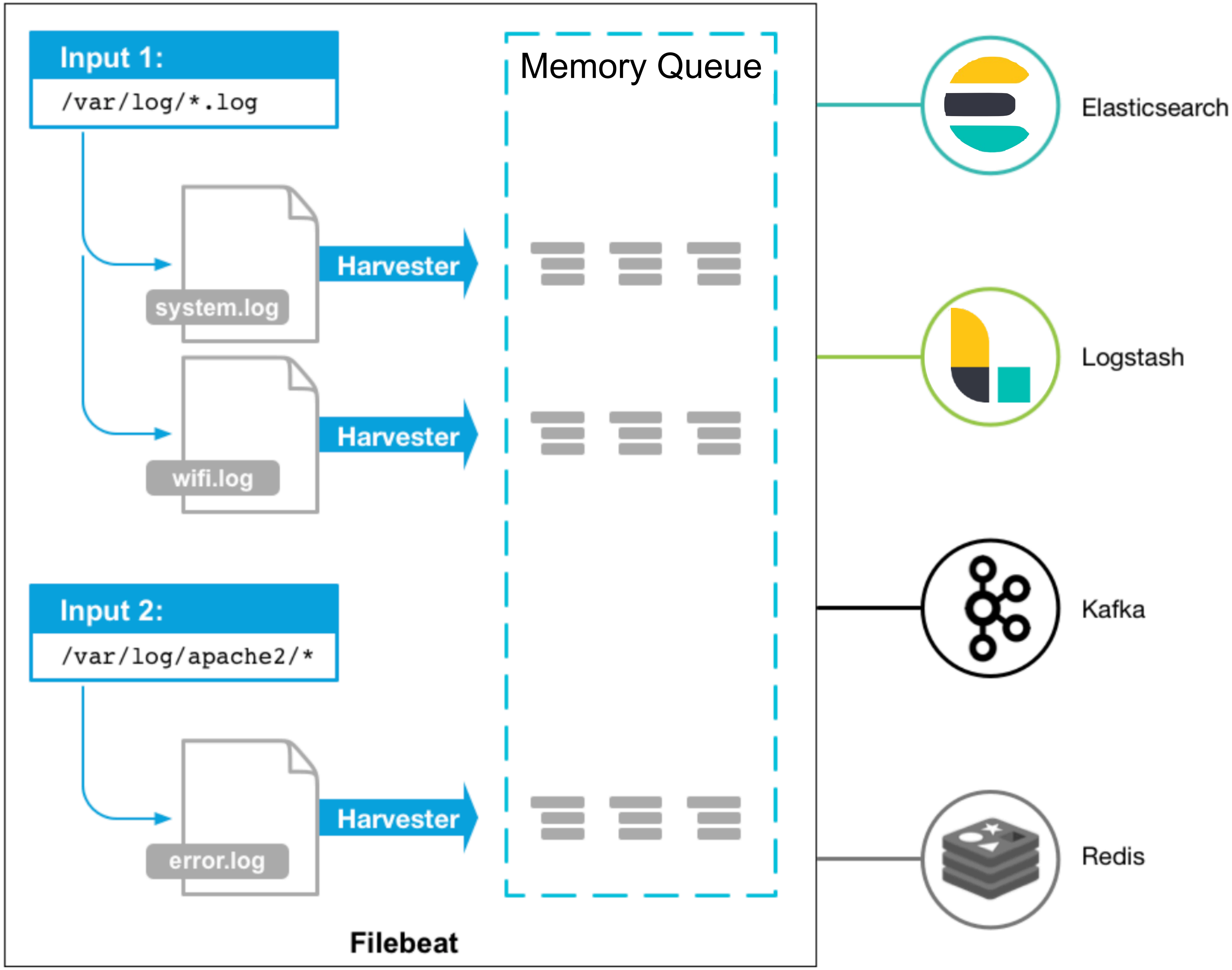
It is a lightweight shipper for forwarding and centralizing log data.

It runs as a binary because it is written in Go, so no runtime library is needed.

It is easy to deploy across many architectures.

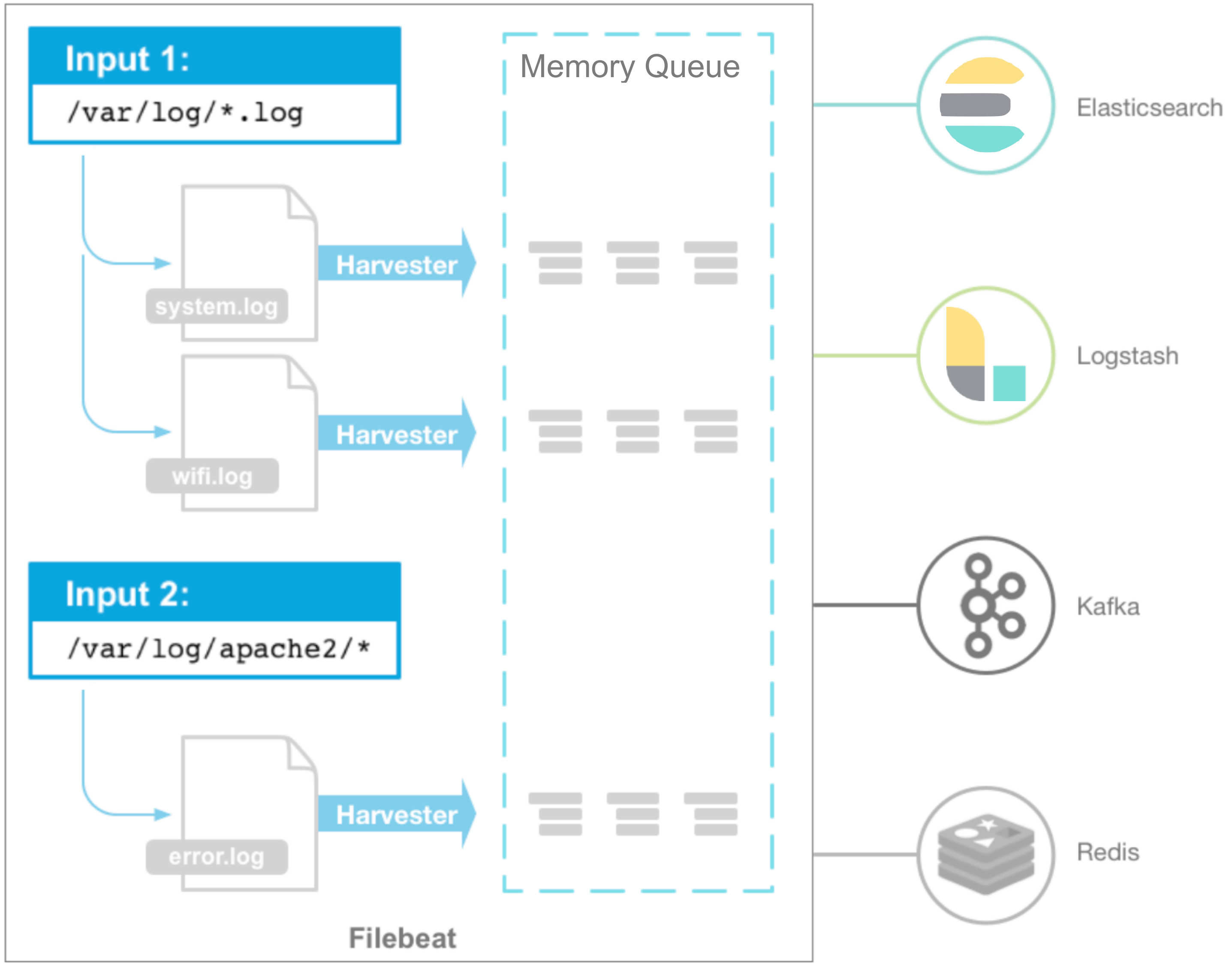
It is possible to scale your data shippers independently of your processors.

Filebeat Architecture



Inputs

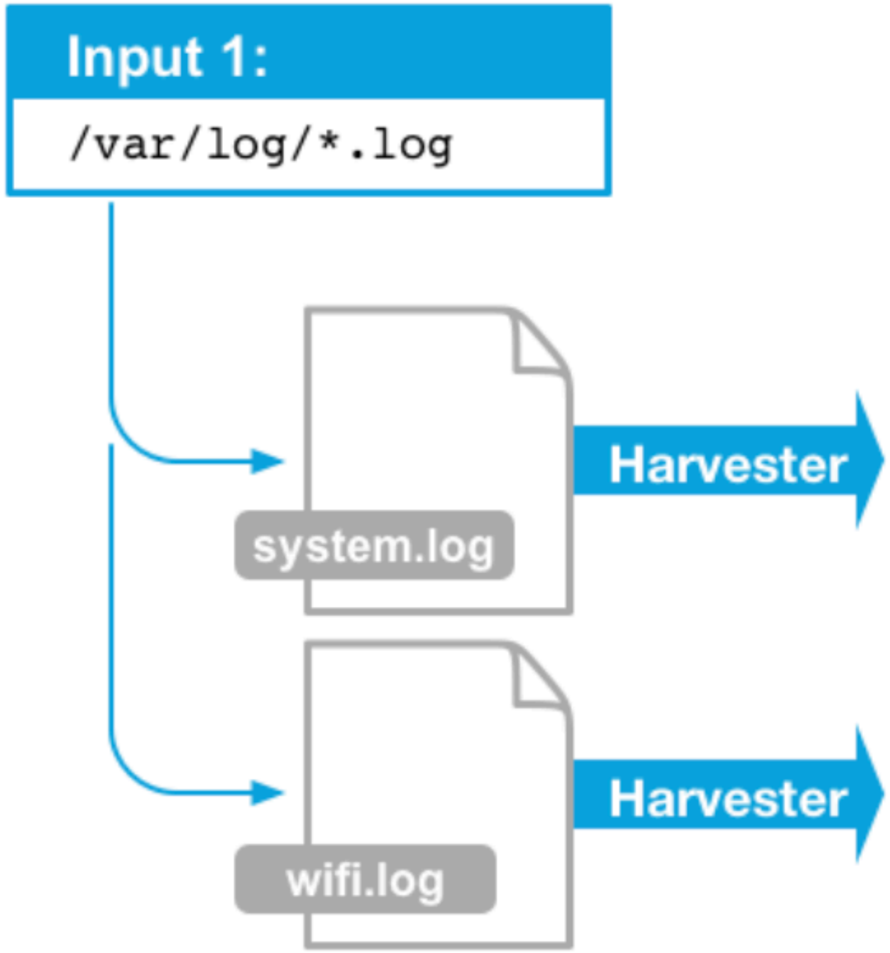
- Each instance of Filebeat can be configured with one or more ***inputs***
- Each input can be configured to monitor one or more file paths



Harvesters

- For each file that an input locates, Filebeat starts a ***harvester***
- Each harvester reads a single file for new log data

The new log data is aggregated and sent to the configured output by libbeat.



Filebeat Configuration

- Default configuration file is **filebeat.yml**
- Sample configuration

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/*.log
    #- c:\programdata\elasticsearch\logs\*

output.elasticsearch:
  hosts: ["localhost:9200"]

setup.kibana:
  host: "localhost:5601"
```

There is also a full example configuration file called **filebeat.reference.yml** that shows all non-deprecated options.

Multiple Inputs

- You can specify multiple inputs
  - and you can specify the same input type more than once

```
filebeat.inputs:
- type: log
  paths:
    - /var/log/system.log
    - /var/log/wifi.log
- type: log
  paths:
    - "/var/log/apache2/*"
  fields:
    apache: true
  fields_under_root: true
```

- Available input types
  - https://www.elastic.co/guide/en/beats/filebeat/current/configuration-filebeat-options.html

Configuration Options

- Filebeat inputs support many configurations options
- For example, the **ignore_older** is an interesting one
    - when enabled Filebeat ignores any files that were modified before the specified timespan
    - it is useful if you keep log files for a long time
    - you can use time strings like 2h (2 hours) and 5m (5 minutes)
    - the default is 0 (which means that it is **disabled** by default)

```
filebeat.inputs:
- type: log
  ...
  ignore_older: 24h
```

Filtering Data

- Filebeat also allows you to use regular expressions to filter exported data

```
filebeat.inputs:
- type: log
  paths:
    - /var/log/myapp/*.log
  include_lines: ['^ERR', '^WARN']
```

export any lines that start with "ERR" or "WARN"

```
filebeat.inputs:
- type: log
  paths:
    - /var/log/myapp/*.log
  exclude_lines: ['^DBG']
```

drop any lines that start with "DBG"

```
filebeat.inputs:
- type: log
  paths:
    - /var/log/myapp/*
  exclude_files: ['\.gz$']
```

ignore all the files that have a `gz` extension

Configuring the Output

- Filebeat supports the following outputs
  - Elasticsearch
  - Logstash
  - Kafka
  - Redis
  - File
  - Console
  - Cloud
- But typically you either send events directly to Elasticsearch
  - or to Logstash for additional processing

Configuring the Elasticsearch Output

- When you specify Elasticsearch for the output
  - Filebeat sends the transactions directly to Elasticsearch
  - Filebeat uses the Elasticsearch HTTP API to send data
- Configuration example

```
output.elasticsearch:
   hosts: ["https//localhost:9200"]
   username: "filebeat_internal"
   password: "YOUR_Password"
```

If Elasticsearch is secured, set credentials before you run the commands that set up and start Filebeat.

Configuring the Cloud Output

- Filebeat comes with two settings that simplify the **cloud** output configuration

```
cloud.id: "staging:kdjfsljliejdsklyuiuoejoiujk"
cloud.auth "elastic:YOUR_PASSWORD"
```

- The Cloud ID is used by Filebeat to resolve Elasticsearch and Kibana URLs
  - overriding **output.elsaticsearch.hosts** and **setup.kibana.host**
- The Cloud auth credentials authenticate Filebeat
  - overriding **output.elasticsearch.username** and **output.elasticsearch.password**
  - it can also be used to set the **setup.kibana.username** and **setup.kibana.password** options

Configuring Template Loading

- Filebeat automatically loads the recommended template
  - which is configured in **fields.yml**
- You can change the default configurations
  - load a different template

```
setup.template.name: "your_template_name"
setup.template.fields: "path/to/fields.yml"
```

  - overwrite an existing template

```
setup.template.overwrite: true
```

If the template already exists, it’s not overwritten unless you configure Filebeat to do so.

  - disable automatic template loading

```
setup.template.enabled: false
```

If you disable automatic template loading, you need to load the template manually.

Changing the Index Name

- The index name is set by **output.elasticsearch.index**
  - the default is **filebeat-%{[agent.version]}-%{+yyyy.MM.dd}**
  - e.g. **filebeat-7.3.1-2019-10-08**
- The index setting is ignored when Index Lifecycle Management is enabled
  - starting with version 7.0, Filebeat uses ILM by default
- If you want to change the index setting
  - you also need to configure the **setup.template.name** and **setup.template.pattern** options
  - and disable ILM if it is enabled

```
setup.ilm.enabled: false
```

Testing Configurations

- You can use the **test** command to test your configurations
  - after you define them
- Testing the configuration settings

```
./filebeat test config
```

- Testing the output configuration

```
./filebeat test output
```

- This commands will test **filebeat.yml**
  - you can use the **-c** option to test a specific file

```
./filebeat test -c specific.yml config
./filebeat test -c specific.yml output
```

Getting Started with Filebeat

1. Install Filebeat

2. Configure Filebeat

3. Setup Elasticsearch index templates, Kibana dashboards, machine learning job configurations and ingest pipelines

```
./filebeat setup
```

Make sure Elasticsearch and Kibana are reachable.

4. Start Filebeat

```
./filebeat -e
```

The **-e** flag is optional and sends output to standard error instead of syslog.

5. View the sample Kibana dashboards

Lesson 1

Review - Filebeat Architecture

elastic

Summary

- Filebeat is a light weight shipper to send logs to Elasticsearch or other sources
- Filebeat is made up of Inputs and Harvesters
- Inputs can be configured to monitor one or more file paths
- Harvesters read a single file for new content
- You can include and exclude lines
- You can exclude different files
- The environment setup only needs to be run once

Quiz

1. **True** or **False**: Filebeat is made up of Inputs and Harvesters.
2. **True** or **False**: Inputs can have multiple paths.
3. What would you have to add to your **filebeat.yml** file to allow for excluding lines starting with **"Info"**?
4. What would you have to add to your **filebeat.yml** file to allow to exclude all **.tar.gz** files?

Lesson 1

Lab - Filebeat Architecture

Lesson 2

Modules

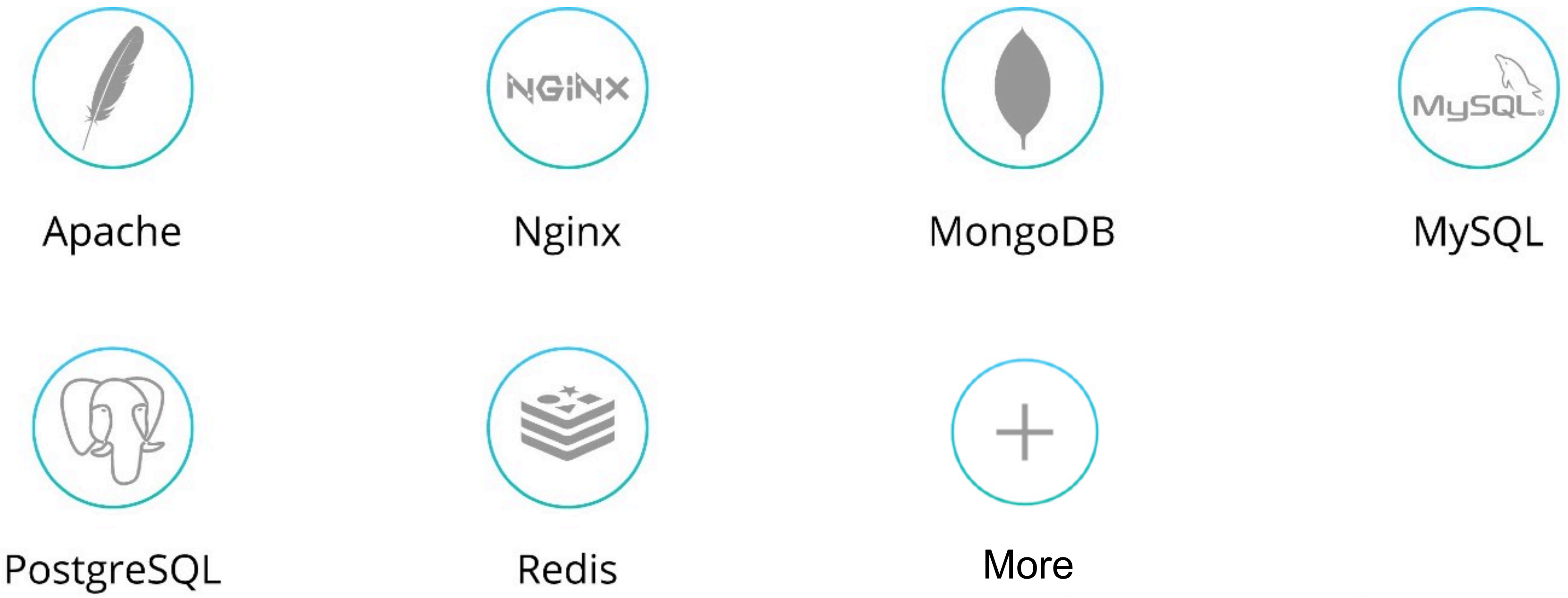
elastic

Modules Overview

- Filebeat modules **simplify** the collection, parsing and visualization of common log formats
- How does it **simplify** this?
  - the Filebeat input configurations contain the default paths where to look for the log files
  - the Elasticsearch ingest node pipeline definitions are used to parse the log lines
  - the field definitions are used to configure Elasticsearch with the correct types for each field
  - the sample Kibana dashboards, when available, can be used to visualize the log files

Supported Modules

- Filebeat provides a set of pre-built modules
  - which you can use to rapidly implement a log monitoring solution
  - complete with sample dashboards and data visualizations
- These modules support common log formats such as



Apache

Nginx

MongoDB

MySQL

PostgreSQL

Redis

More

- https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-modules.html

Deeper Look Into One Module Example

- The **nginx** module parses access and error logs created by the NGINX HTTP server
- It performs a few tasks under the hood
  - sets the default paths to the log files (which you can change)
  - makes sure each multi-line log event gets sent as a single event
  - uses ingest node to parse and process the log lines
  - shapes the data into a structure suitable for visualizing in Kibana
  - deploys dashboards for visualizing the log data
- The **nginx** module was tested with logs from version 1.10

Prerequisites

- Before running Filebeat modules

  1. Install and configure the Elastic Stack
  2. Install Filebeat
  3. Verify that Elasticsearch and Kibana are running
  4. Verify that Elasticsearch is ready to receive data from Filebeat

Enabling Modules

- You can use the **modules** command to **enable** a module

```
./filebeat modules enable nginx
```

- You can also enable more than one module at a time

```
./filebeat modules enable nginx system mysql
```

- And you can see the list of enabled and disabled modules

```
./filebeat modules list
```

- To disable modules you can use the **disable** command

Other Ways to Enable Modules

- By default modules are enabled in the **modules.d** directory

```
./filebeat modules enable nginx system mysql
```

- You can also enable modules when you run Filebeat

```
./filebeat --modules nginx,system,mysql
```

Works well when you are getting started.

- And you can also enabled modules in the **filebeat.yml** file

```
filebeat.modules:
- module: nginx
- module: system
- module: mysql
```

It is a practical approach if you have upgraded from a previous version of Filebeat.

Configuring Modules

- You can refine the behavior of Filebeat modules through their configuration files
- The directory **modules.d** include one configuration file for each supported module

```
[elastic@server1 modules.d]$ ls
apache.yml.disabled      auditd.yml.disabled        cisco.yml.disabled
coredns.yml.disabled     elasticsearch.yml.disabled envoyproxy.yml.disabled
googlecloud.yml.disabled haproxy.yml.disabled       icinga.yml.disabled
iis.yml.disabled         iptables.yml.disabled      kafka.yml.disabled
kibana.yml.disabled      logstash.yml.disabled      mongodb.yml.disabled
mssql.yml.disabled       mysql.yml                  nats.yml.disabled
netflow.yml.disabled     nginx.yml                  osquery.yml.disabled
panw.yml.disabled        postgresql.yml.disabled.   rabbitmq.yml.disabled
redis.yml.disabled       santa.yml.disabled.        suricata.yml.disabled
system.yml               traefik.yml.disabled.      zeek.yml.disabled
```

Enabled modules have configuration files that do not end with the **.disabled** extension.

Configuration Example

- The following example shows how to set paths in the **modules.d/nginx.yml** file to override the default paths

```
- module: nginx
  access:
    enabled: true
    var.paths: ["/path/to/log/nginx/access.log*"]
  error:
    enabled: true
    var.paths: ["/path/to/log/nginx/error.log*"]
```

- You can specify the same settings through command line

```
./filebeat --modules nginx \
-M "nginx.access.var.paths=[/path/to/log/nginx/access.log*]" \
-M "nginx.error.var.paths=[/path/to/log/nginx/error.log*]"
```

Advanced Settings

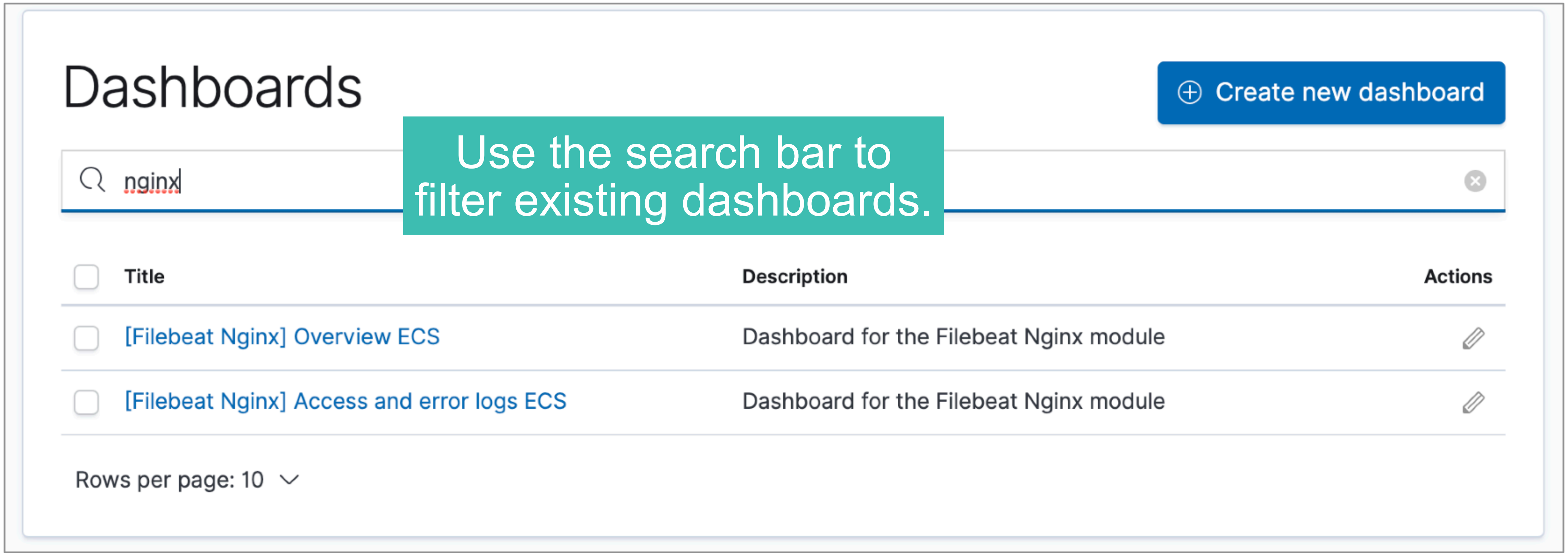
- Behind the scenes, each module starts a Filebeat input
- You can add or override any input settings
- For example, you can set **close_eof** to true in the module configuration
  - when this option is enabled
  - Filebeat closes a file as soon as the end of file is reached

```
- module: nginx
  access:
    input:
      close_eof: true
```

Useful setting when your files are written once and not updated from time to time.

Explore Dashboards

- Open your browser and navigate to the Dashboards app



Dashboards

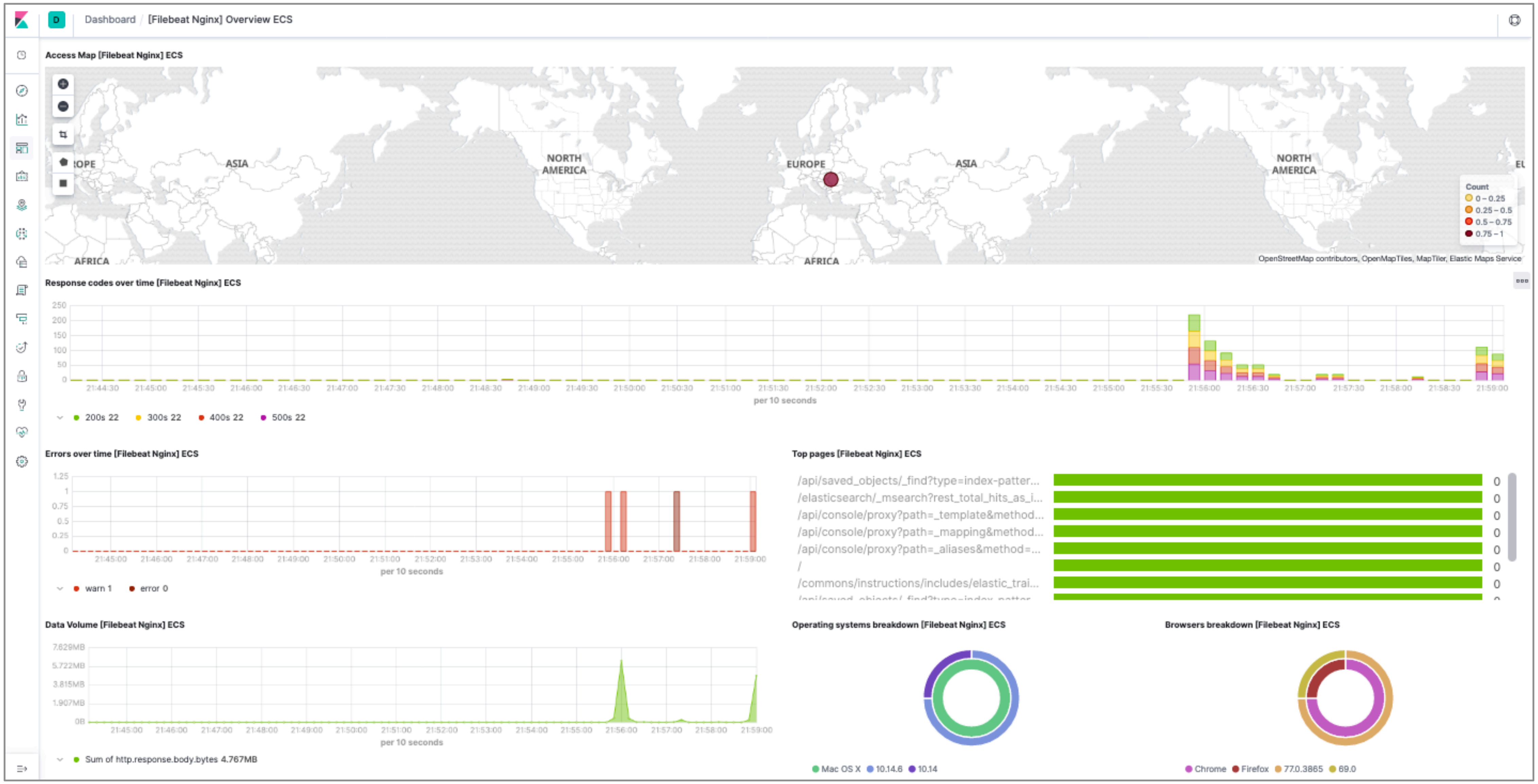
Use the search bar to filter existing dashboards.

□ Title	Description	Actions
□ [Filebeat Nginx] Overview ECS	Dashboard for the Filebeat Nginx module	✎
□ [Filebeat Nginx] Access and error logs ECS	Dashboard for the Filebeat Nginx module	✎

Rows per page: 10

Dashboard Example

- Open a dashboard and explore the visualizations for your parsed logs


Dashboard / [Filebeat Nginx] Overview ECS

Access Map [Filebeat Nginx] ECS

OpenStreetMap contributors, OpenMapTiles, MapTiler, Elastic Maps Service

Response codes over time [Filebeat Nginx] ECS

Errors over time [Filebeat Nginx] ECS

Top pages [Filebeat Nginx] ECS

/api/saved_objects/_find?type=index-patter... 0
/elasticsearch/_msearch?rest_total_hits_as_i... 0
/api/console/proxy?path=_template&method... 0
/api/console/proxy?path=_mapping&method... 0
/api/console/proxy?path=_aliases&method=... 0
/ 0
/commons/instructions/includes/elastic_trai... 0

Data Volume [Filebeat Nginx] ECS

Operating systems breakdown [Filebeat Nginx] ECS

Browsers breakdown [Filebeat Nginx] ECS

Lesson 2

Review - Modules

Summary

- Filebeat comes with many pre-built modules
- Modules are designed to simplify the ingestion of common log files into Elasticsearch
- Modules must be enabled
- Most modules come with pre-built dashboards
- You can edit the yaml file to define different paths to logs

Quiz

1. **True** or **False**: You can only enabled one module at a time.
2. **True** or **False**: Filebeat comes with prebuilt modules for most common log files.
3. **True** or **False**: You can configure the module's .yml file to change the location of where Filebeat looks for log files.

Lesson 2

Lab - Modules

Lesson 3

Resilience

Checking for New Files

- How often Filebeat checks for new files?
  - this is specified by the **scan_frequency** setting
  - which is set to **10s** by default
  - but you can set it to **1s** for scanning as frequently as possible
  - it is **not** recommended to set it **<1s**

Checking for New Lines

- How often Filebeat checks for new lines in a file?
  - the backoff options specify how aggressively Filebeat crawls open files for updates
- The **backoff** option defines how long Filebeat waits before checking a file again after **EOF** is reached
  - the default is **1s**
- The **backoff_factor** option specifies the factor used to exponentially increment **max_backoff** when no new entries are found
  - the default is **2**
- The **max_backoff** option defines the maximum time to wait before checking a file again after **EOF** is reached
  - the default is **10s**

Recovering

- What happens if you stop Filebeat?
  - Filebeat uses a registry path to track progress
  - so it can use it to get back from where it stopped when restarted
- What does the registry stores?
  - current log files being parsed
  - offset into each log file
  - inode and device information (Linux file systems)
- What do you do if you want to import the same data again?
  - just delete the registry
  - but be careful deleting it in production

Where is this Registry?

- It is located in **${path.data}/registry**
  - **data/registry** for **.tar.gz** archives
  - **/var/lib/filebeat/registry** for DEB and RPM packages
  - **C:\ProgramData\filebeat\registry** for the Windows zip file
- You can use the **filebeat.registry.path** setting to define another location
- Use the **filebeat.registry.migrate_file** to migrate old registry files to the new directory format
  - if migrating from Filebeat 6.x to 7.x

Troubleshooting

Is Log Data Being Sent?

- You can check Filebeat logs whenever you need to confirm whether log data is being sent
  - review the Filebeat directory layout to see where they are stored

Type	Description	Location
`home`	Home of the Filebeat installation.	`{extract.path}`
`bin`	The location for the binary files.	`{path.home}/bin`
`config`	The location for configuration files.	`{path.home}`
`data`	The location for persistent data files.	`{path.home}/data`
`logs`	The location for the logs created by Filebeat.	`{path.home}/logs`

Debug Options

- You can increase the verbosity of debug messages
- You can do that by enabling one or more debug selectors
- For example, you can view the published transactions

```
./filebeat -e -d "publish"
```

- As another example, you can view all the debugging output

```
./filebeat -e -d "*"
```

Fair warning,
it is quite a lot.

Common Problems

elastic

Too Many Open File Handlers

- If Filebeat is harvesting a large number of files
  - then the number of open files can become an issue
- If a file is updated after the harvester is closed
  - it will be picked up again after **scan_frequency** has elapsed
- However, if the file is moved or deleted while the harvester is closed
  - then Filebeat will not be able to pick up the file again
  - and any data that the harvester hasn't read will be lost
- You can use **close_*** configuration options to close the harvester after a certain criteria or time
  - it is helpful to close files that are no longer active

Registry File Is Too Large

- If a large number of new lines are produced every day
  - then the registry might grow to be too large
- To reduce registry size there are two configuration options
  - use **clean_inactive** for old files that you no longer touch
  - use **clean_removed** for old files that are removed from disk

Inode Reuse Causes Filebeat to Skip Lines

- On Linux file systems, Filebeat uses inode and device information to identify files
  - when a file is removed, the inode may be assigned to a new file
  - Filebeat assumes the new file is the same as the old one
  - and tries to continue reading at the old position
- You can use **clean_*** options to clean up state entries in the registry
  - thus preventing a potential inode reuse issue
  - besides reducing the registry size

Lesson 3

Review - Resilience

Summary

- The **scan_frequency** runs every 10 seconds to help pick up new files
- Using backoff options offers Filebeat the opportunity to back off if the file handler is too busy
- All information is held in the registry and can be helpful for holding state
- You can use the **-d** command line option to increase the verbosity of debug messages
- The **close_*** and **clean_*** options help closing inactive file handlers and controlling registry size, respectively

Quiz

1. **True** or **False**: The **scan_frequency** option is set to 10 seconds by default?
2. **True** or **False**: It is a good idea to set **scan_frequency** lower than 1 second to scan as frequently as possible.
3. **True** or **False**: Filebeat uses a registry to keep track of the files that it has ingested.

Lesson 3

Lab - Resilience

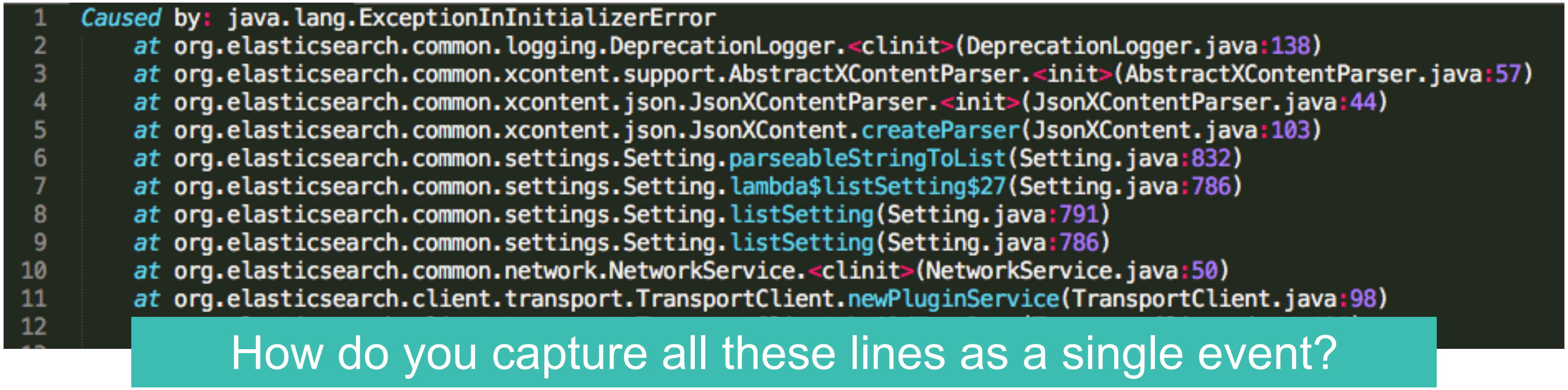
elastic

Lesson 4

Multiline Processing

Multiline Events

- Multiline events must be collected at the source
  - otherwise order is not guaranteed
- Uses regular expressions to define start or end line of the event



```
Caused by: java.lang.ExceptionInInitializerError
    at org.elasticsearch.common.logging.DeprecationLogger.<clinit>(DeprecationLogger.java:138)
    at org.elasticsearch.common.xcontent.support.AbstractXContentParser.<init>(AbstractXContentParser.java:57)
    at org.elasticsearch.common.xcontent.json.JsonXContentParser.<init>(JsonXContentParser.java:44)
    at org.elasticsearch.common.xcontent.json.JsonXContent.createParser(JsonXContent.java:103)
    at org.elasticsearch.common.settings.Setting.parseableStringToList(Setting.java:832)
    at org.elasticsearch.common.settings.Setting.lambda$listSetting$27(Setting.java:786)
    at org.elasticsearch.common.settings.Setting.listSetting(Setting.java:791)
    at org.elasticsearch.common.settings.Setting.listSetting(Setting.java:786)
    at org.elasticsearch.common.network.NetworkService.<clinit>(NetworkService.java:50)
    at org.elasticsearch.client.transport.TransportClient.newPluginService(TransportClient.java:98)
```

How do you capture all these lines as a single event?

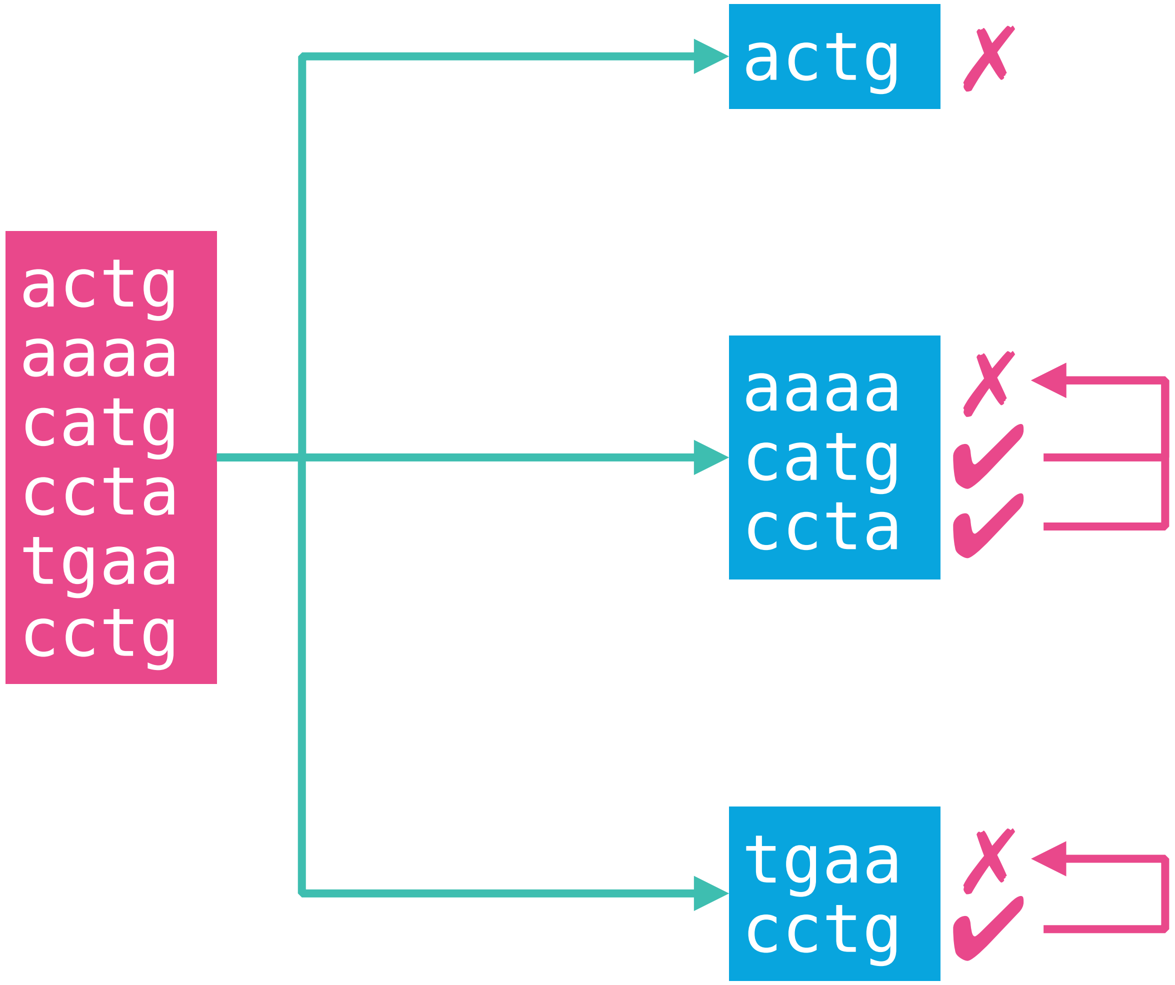
Multiline Event Settings

- The **multiline.pattern** setting
  - regular expression pattern to match
- The **multiline.negate** setting
  - whether the pattern is negated
  - the default is **false**
- The **multiline.match** setting
  - how to combine matching lines into an event
  - available settings are **after** or **before**

How Negate and Match Work Together

- Consecutive lines that match the pattern are appended to the previous line that doesn't match

```
filebeat.inputs:
   multiline.pattern: '^c'
   multiline.negate:  false
   multiline.match:   after
```



actg
aaaa
catg
ccta
tgaa
cctg

actg ✗

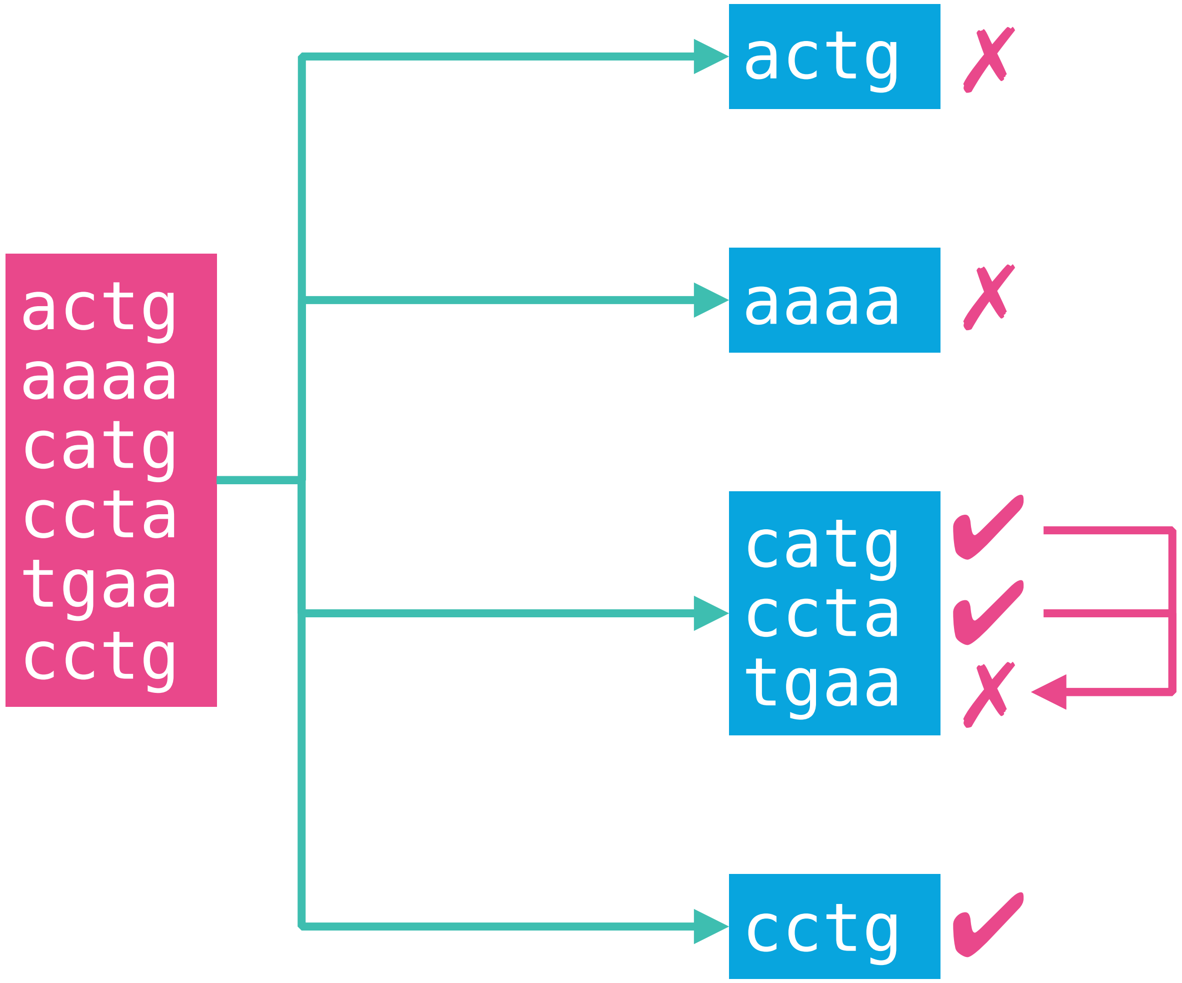
aaaa ✗
catg ✓
ccta ✓

tgaa ✗
cctg ✓

How Negate and Match Work Together

- Consecutive lines that match the pattern are prepended to the next line that doesn't match

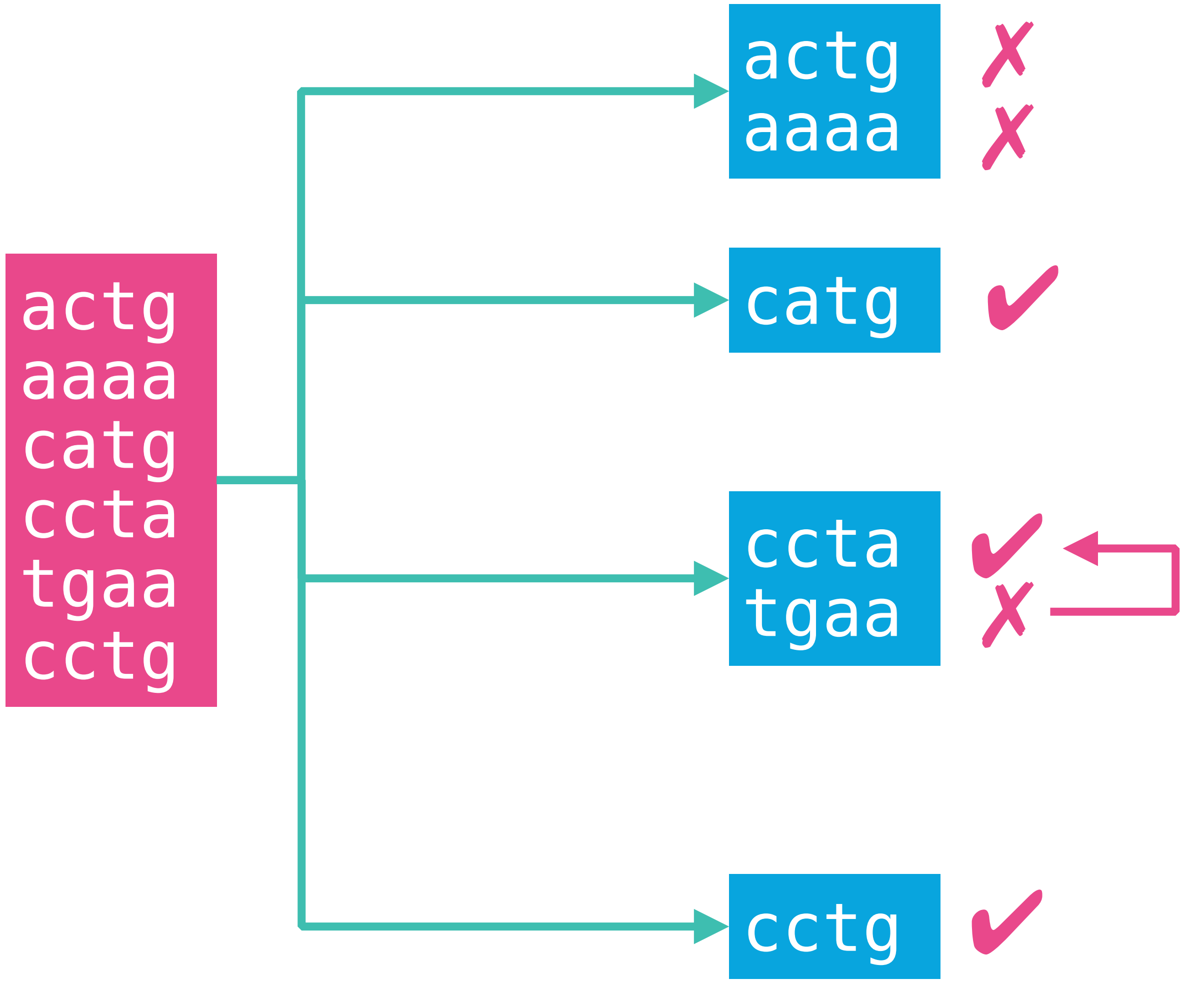
```
filebeat.inputs:
   multiline.pattern: '^c'
   multiline.negate:  false
   multiline.match:   before
```



How Negate and Match Work Together

- Consecutive lines that don't match the pattern are appended to the previous line that does match

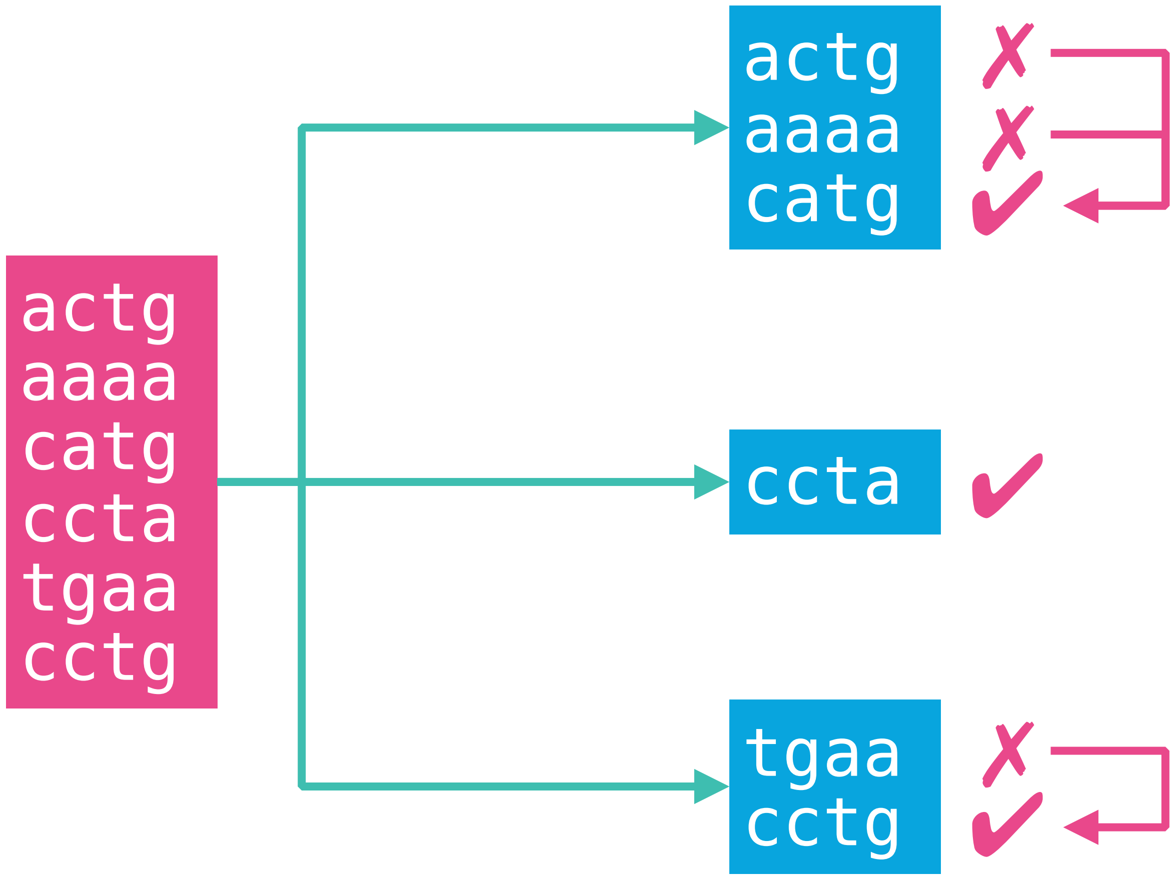
```
filebeat.inputs:
   multiline.pattern: '^c'
   multiline.negate:  true
   multiline.match:   after
```



How Negate and Match Work Together

- Consecutive lines that don’t match the pattern are prepended to the next line that does match

```
filebeat.inputs:
   multiline.pattern: '^c'
   multiline.negate:  true
   multiline.match:   before
```



actg
aaaa
catg
ccta
tgaa
cctg

actg ✗
aaaa ✗
catg ✓

ccta ✓

tgaa ✗
cctg ✓

Multiline Event Settings Example

- Defined in the inputs section of the configuration

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/*.log

  multiline.pattern: '^\['
  multiline.negate: true
  multiline.match: after
```

any line not starting with **[** belongs to the previous line that does

Java Stack Trace

- Java stack traces consist of multiple lines

```
Exception in thread "main" java.lang.NullPointerException
        at com.example.myproject.Book.getTitle(Book.java:16)
        at com.example.myproject.Author.getBookTitles(Author.java:25)
        at com.example.myproject.Bootstrap.main(Bootstrap.java:14)
```

- Note that **after** the initial line each line starts with **whitespace**
- It is possible to parse these multiline events as follows

```
multiline.pattern: '^[[:space:]]'
multiline.negate: false
multiline.match: after
```

Complex Java Stack Trace

- Java stack traces can be slightly more complicated

```
Exception in thread "main" java.lang.IllegalStateException: A book has a null property
       at com.example.myproject.Author.getBookIds(Author.java:38)
       at com.example.myproject.Bootstrap.main(Bootstrap.java:14)
Caused by: java.lang.NullPointerException
       at com.example.myproject.Book.getId(Book.java:22)
       at com.example.myproject.Author.getBookIds(Author.java:35)
       ... 1 more
```

- It is possible to parse these multiline events as follows

```
multiline.pattern: '^[[:space:]]+(at|\.{3})\b|^Caused by:'
multiline.negate: false
multiline.match: after
```

a line that begins with **spaces** followed by the word **at** or ...
a line that begins with the words **Caused by:**

Timestamped Multiline Events

- Activity logs from services typically begin with a timestamp
  - followed by information on the specific activity

```
[2015-08-24 11:49:14,389][INFO ][env] [Letha] using [1] data paths,
mounts [[/(/dev/disk1)]], net usable_space [34.5gb],
net total_space [118.9gb], types [hfs]
```

- It is possible to parse these multiline events as follows

```
multiline.pattern: '^\[[0-9]{4}-[0-9]{2}-[0-9]{2}'
multiline.negate: true
multiline.match: after
```

This configuration uses the **negate: true** and **match: after** settings to specify that any line that does not match the specified pattern belongs to the previous line.

More Multiline Event Settings

- The **multiline.flush_pattern** setting
  - a regular expression to flush the current multiline from memory
  - ends the multiline-message
- The **multiline.max_lines** setting
  - maximum number of lines that can be combined into one event
  - this means that additional lines are discarded
  - the default is **500**
- The **multiline.timeout** setting
  - Filebeat sends the multiline event after the specified timeout
  - even if no new pattern is found to start a new event
  - the default is **5s**

Multiline Application Logs

- Sometimes your application logs contain events which start and end with custom markers

```
[2015-08-24 11:49:14,389] Start new event
[2015-08-24 11:49:14,395] Content of processing something
[2015-08-24 11:49:14,399] End event
```

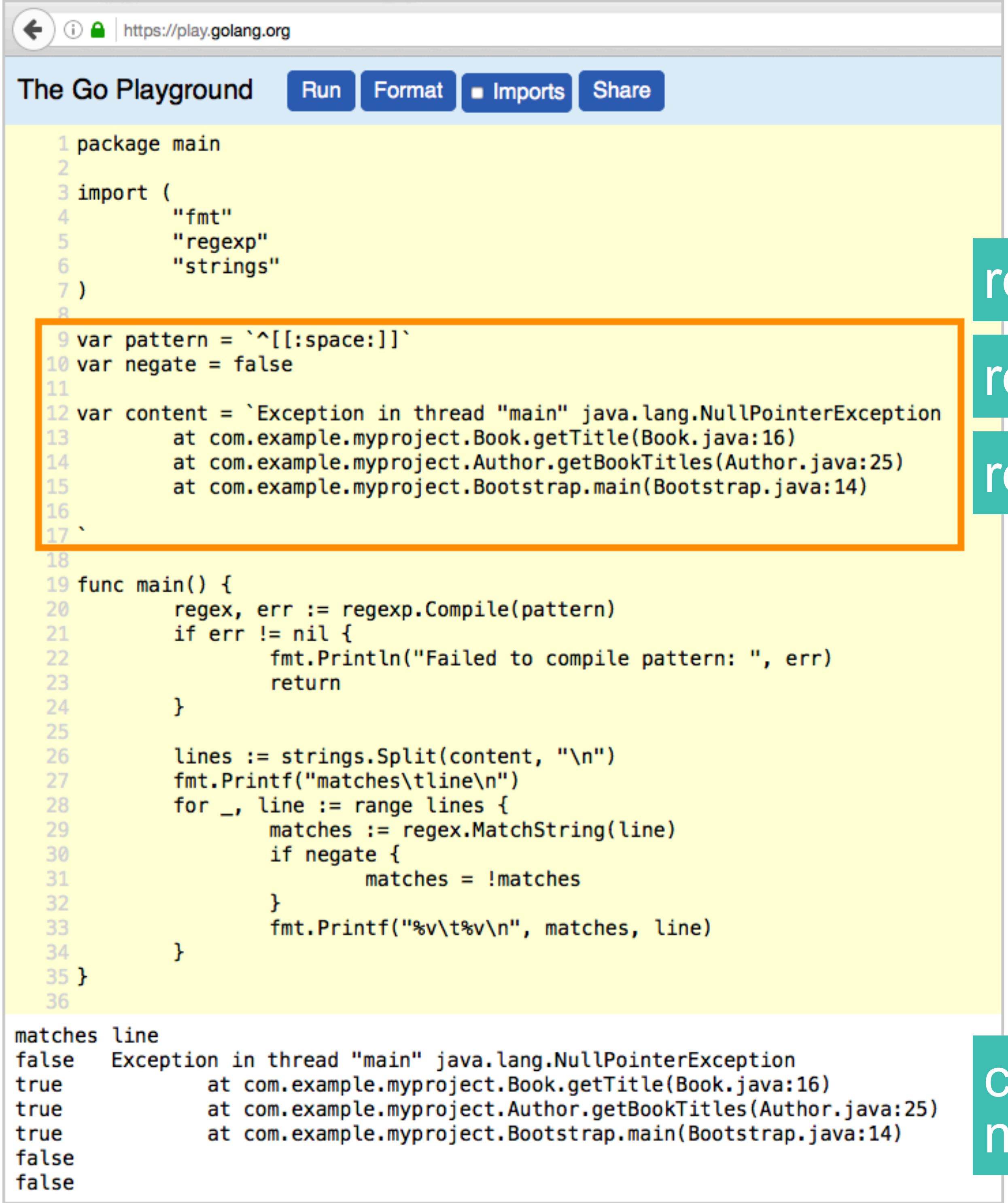
- It is possible to parse these multiline events as follows

```
multiline.pattern: 'Start new event'
multiline.negate: true
multiline.match: after
multiline.flush_pattern: 'End event'
```

Note how the example uses **multiline.flush_pattern** to flush the multiline event, ending the multiline-message.

How To Test Multiline Patterns?

- https://play.golang.org/p/uAd5XHxscu



```go
package main

import (
        "fmt"
        "regexp"
        "strings"
)

var pattern = `^[[:space:]]`
var negate = false

var content = `Exception in thread "main" java.lang.NullPointerException
        at com.example.myproject.Book.getTitle(Book.java:16)
        at com.example.myproject.Author.getBookTitles(Author.java:25)
        at com.example.myproject.Bootstrap.main(Bootstrap.java:14)

`

func main() {
        regex, err := regexp.Compile(pattern)
        if err != nil {
                fmt.Println("Failed to compile pattern: ", err)
                return
        }

        lines := strings.Split(content, "\n")
        fmt.Printf("matches\tline\n")
        for _, line := range lines {
                matches := regex.MatchString(line)
                if negate {
                        matches = !matches
                }
                fmt.Printf("%v\t%v\n", matches, line)
        }
}
```

```
matches line
false   Exception in thread "main" java.lang.NullPointerException
true            at com.example.myproject.Book.getTitle(Book.java:16)
true            at com.example.myproject.Author.getBookTitles(Author.java:25)
true            at com.example.myproject.Bootstrap.main(Bootstrap.java:14)
false
false
```

replace **pattern** with your **multiline.pattern**

replace **negate** with your **multiline.negate**

replace **content** with one message example

click on Run and check which lines in the message match your specified configuration

Lesson 4

Review - Multiline Processing

Summary

- Some log files have multiline events
- Filebeat uses regular expression patterns to match the beginning line of a multiline log message
- The **multiline.pattern** setting specify what is the regular expression pattern to match
- Using **multiline.negate** and **multiline.match** you can pair up the beginning and ending of your multiline event
- You can also use **multiline.flush_pattern** to signal the end of a multiline event

Quiz

1. **True** or **False**: The **multiline.pattern** setting uses regular expressions to determine the first char in a new line?
2. What is the difference between match **before** and match **after**?
3. **True** or **False**: You can use **multiline.flush_pattern** to signal the beginning of a multiline event.

Lesson 4

Lab - Multiline Processing

Quiz Answers

Filebeat Architecture

1. True
2. True
3. exclude_lines: ['^Info']
4. exclude_files: ['\.tar.gz']

Modules

1. False

2. True

3. True

Resilience

1. True
2. False - It is a bad practice to set the scan_frequency lower then 1 second
3. True

Multiline Processing

1. True
2. Match before will associate all lines to the pervious one, until another match is made. The Match after will associate all lines until a match is made.
3. False