input {

beats {

port => 5044

ssl => true

ssl\_certificate\_authorities => ["/etc/logstash/certs/http\_ca.crt"]

ssl\_certificate => "/etc/logstash/certs/logstash.crt"

ssl\_key => "/etc/logstash/certs/logstash.key"

ssl\_verify\_mode => "force\_peer"

}

}

filter {

if [event][module] == "apache" {

if [fileset][name] == "access" {

grok {

match => { "message" => ["%{IPORHOST:[source][address]} - %{DATA:[user][name]} \[%{HTTPDATE:[apache][access][time]}\] \"(?:%{WORD:[http][request][method]} %{DATA:[url][original]} HTTP/%{NUMBER:[http][version]:float}|-)?\" %{NUMBER:[http][response][status\_code]:int} (?:%{NUMBER:[http][response][body][bytes]:int}|-)( \"%{DATA:[http][request][referrer]}\")?( \"%{DATA:[user\_agent][original]}\")?","%{IPORHOST:[source][address]} - %{DATA:[user][name]} \[%{HTTPDATE:[apache][access][time]}\] \"-\" %{NUMBER:[http][response][status\_code]:int} -","\[%{HTTPDATE:[apache][access][time]}\] %{IPORHOST:[source][address]} %{DATA:[apache][access][ssl][protocol]} %{DATA:[apache][access][ssl][cipher]} \"%{WORD:[http][request][method]} %{DATA:[url][original]} HTTP/%{NUMBER:[http][version]:float}\" %{NUMBER:[http][response][body][bytes]:int}"

] }

remove\_field => [ "message" ]

}

grok {

match => { "[source][address]" => ["^(%{IP:[source][ip]}|{HOSTNAME:[source][domain]})$"]}

}

mutate {

rename => { "[event][created]" => "@timestamp" }

}

date {

match => [ "[apache][access][time]", "dd/MMM/yyyy:H:m:s Z" ]

remove\_field => "[apache][access][time]"

}

useragent {

source => "[user\_agent][original]"

}

geoip {

source => "[source][ip]"

target => "[source][geo]"

}

}

else if [fileset][name] == "error" {

grok {

match => { "message" => ["\[%{APACHE\_TIME:[apache][error][timestamp]}\] \[%{LOGLEVEL:[log][level]}\]( \[client %{IPORHOST:[source][address]}(:%{POSINT:[source][port]:int})?\])? %{GREEDYDATA:message}","\[%{APACHE\_TIME:[apache][error][timestamp]}\] \[%{DATA:[apache][error][module]}:%{LOGLEVEL:[log][level]}\] \[pid %{NUMBER:[process][pid]:int}(:tid %{NUMBER:[process][thread][id]:int})?\]( \[client %{IPORHOST:[source][address]}(:%{POSINT:[source][port]:int})?\])? %{GREEDYDATA:message}"

] }

pattern\_definitions => {

"APACHE\_TIME" => "%{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{YEAR}"

}

remove\_field => "message"

}

date {

match => [ "[apache][error][timestamp]", "EEE MMM dd H:m:s YYYY", "EEE MMM dd H:m:s.SSSSSS YYYY" ]

remove\_field => "[apache][error][timestamp]"

}

grok {

match => { "[source][address]" => ["^(%{IP:[source][ip]}|%{HOSTNAME:[source][domain]})$"]}

}

geoip {

source => "[source][ip]"

target => "[source][geo]"

}

}

}

date {

match => ["timestamp", "MMM dd HH:mm:ss"]

}

}

output {

if [@metadata][pipeline] {

elasticsearch {

ssl => true

hosts =>["https://157.230.250.174:9200"]

cacert => "/etc/logstash/certs/http\_ca.crt"

user => "logstash\_internal"

password => "123456"

manage\_template => false

index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"

pipeline => "%{[@metadata][pipeline]}"

}

} else {

elasticsearch {

ssl => true

hosts => ["https://157.230.250.174:9200"]

cacert => "/etc/logstash/certs/http\_ca.crt"

user => "logstash\_internal"

password => "123456"

manage\_template => false

index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"

}

}

}