

# ISPConfig Apache2 Logfiles an Graylog-Server schicken

Diese Konfiguration schickt nginx Access- und Errorlogs über GELF an einen Graylog-Server. Der GELF-Input im Graylog sollte natürlich aktiviert sein. Das Pattern-Matching funktioniert leider noch nicht exakt.

/etc/logstash/patterns.d/apache.conf

```
# get hostname from access.log path
APACHE_WEBSITE /[^\s]+/[^\s]+/[^\s]+/[^\s]+/(?<website>[^\s]+)/

# error
APACHE_ERROR_TIME %{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{YEAR}
APACHE_ERROR_LOG \[%{APACHE_ERROR_TIME:timestamp}\] \[%{LOGLEVEL:loglevel}\] (?:\[client %
```

/etc/logstash/conf.d/apache.conf

```
# apache log input
input {
  file {
    type => "apache-access"
    path => ["/var/log/ispconfig/httpd/*/access.log"]
  }
  file {
    type => "apache-error"
    path => ["/var/log/ispconfig/httpd/*/error.log"]
  }
}

# filters
filter {
  if [type] == "apache-access" {
    grok {
      match => { "message" => "%{COMBINEDAPACHELOG}" }
    }
    grok {
      patterns_dir => [ "/etc/logstash/patterns.d" ]
      match => [ "path", "%{APACHE_WEBSITE}" ]
    }
  }

  if [type] == "apache-error" {
    grok {
      patterns_dir => [ "/etc/logstash/patterns.d" ]
      match => [ "message", "%{APACHE_ERROR_LOG}" ]
    }

    if !("_grokparsefailure" in [tags]) {
      mutate {
        remove_field => [ "message" ]
        add_field => [ "timestamp_submitted", "%{@timestamp}" ]
      }

      date {
        match => [ "timestamp", "EEE MMM dd HH:mm:ss yyyy" ]
        remove_field => [ "timestamp" ]
      }

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

# output
output {
```

```
#stdout {  
#  #codec => "plain"  
#  codec => "rubydebug"  
#}  
gelf {  
  host => "log.myserver.de"  
  port => 12201  
}  
}
```

---

Revision #1

Created 30 April 2021 12:17:19 by magenbrot

Updated 30 April 2021 12:17:54 by magenbrot