FEBRUARY 14, 2024

DevOps Classroom notes 14/Feb/2024

**Logs**

* Apache access logs

192.168.2.20 - - [28/Jul/2006:10:27:10 -0300] "GET /cgi-bin/try/ HTTP/1.0" 200 3395

127.0.0.1 - - [28/Jul/2006:10:22:04 -0300] "GET / HTTP/1.0" 200 2216

127.0.0.1 - - [28/Jul/2006:10:27:32 -0300] "GET /hidden/ HTTP/1.0" 404 7218

* Postgres logs

[2007-08-31 19:22:21.469 ADT] :[unknown] LOG: connection received: host=192.168.2.99 port=52136

[2007-08-31 19:22:21.485 ADT] 192.168.2.99:ossecdb LOG: connection authorized: user=ossec\_user database=ossecdb

[2007-08-31 19:22:22.427 ADT] 192.168.2.99:ossecdb LOG: disconnection: session time: 0:00:00.95 user=ossec\_user database=ossecdb host=192.168.2.99 port=52136

[2007-09-27 11:02:44.941 ADT] 192.168.2.10:ossecdb ERROR: relation "lala" does not exist

[2007-09-27 11:02:46.444 ADT] 192.168.2.10:ossecdb LOG: disconnection: session time: 0:00:35.79 user=ossec\_user database=ossecdb host=192.168.2.10 port=3584

* rabbit mq logs

Dec 26 11:03:04 localhost rabbitmq-server[968]: ## ##

Dec 26 11:03:04 localhost rabbitmq-server[968]: ## ## RabbitMQ 3.11.6. Copyright (c) 2005-2024 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries.

Dec 26 11:03:04 localhost rabbitmq-server[968]: ########## Licensed under the MPL. See https://www.rabbitmq.com/

Dec 26 11:03:04 localhost rabbitmq-server[968]: ###### ##

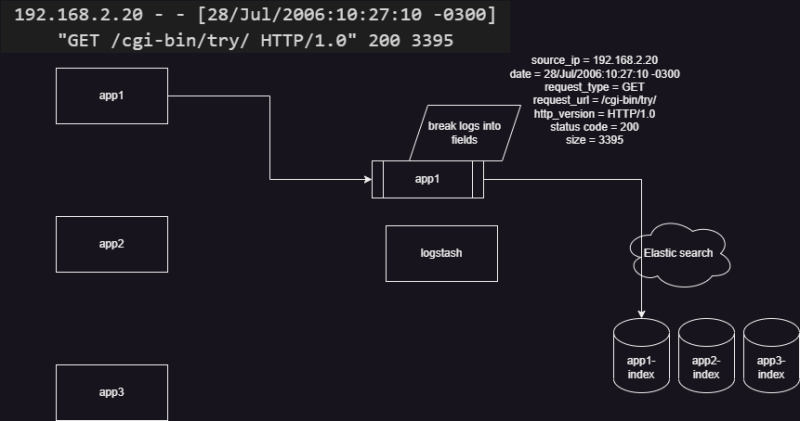
Dec 26 11:03:04 localhost rabbitmq-server[968]: ########## Logs: /var/log/rabbitmq/rabbit@localhost.log

Dec 26 11:03:04 localhost rabbitmq-server[968]: /var/log/rabbitmq/rabbit@localhost\_upgrade.log

Dec 26 11:03:04 localhost rabbitmq-server[968]: Starting broker...

Dec 26 11:03:05 localhost rabbitmq-server[968]: systemd unit for activation check: "rabbitmq-server.service"

Dec 26 11:03:06 localhost rabbitmq-server[968]: completed with 6 plugins.

* To deal with different log formats, elastic stack uses logstash and speaks about creating a log pipeline  
  

**Logstash**

* Logstash is an opensource applciation which transforms logs
* Logstash has 3 stages
  + input stage
    - This accepts inputs from various sources with the help of input plugins [Refer Here](https://www.elastic.co/guide/en/logstash/current/input-plugins.html)
  + filter stage:
    - This stage does the transformations with the help of filter plugins [Refer Here](https://www.elastic.co/guide/en/logstash/current/filter-plugins.html)
  + output stage
    - This stage does load the data to the destination with the help of output plugins [Refer Here](https://www.elastic.co/guide/en/logstash/current/output-plugins.html)
* Pipeline in log stash is represented as shown below

input {

<plugin> {

paramters

}

}

filter {

<plugin> {

paramters

}

}

output {

<plugin> {

paramters

}

}

* filter is optional in this

**Experimenting with logstash**

* Setup logstash on a ubuntu machine (or use a docker container) [Refer Here](https://www.elastic.co/guide/en/logstash/current/installing-logstash.html#_apt)
* We need to create a pipeline [Refer Here](https://www.elastic.co/guide/en/logstash/current/configuration.html)
* Lets create a simple pipeline with a file name basic.conf

input

{

stdin

{

}

}

output

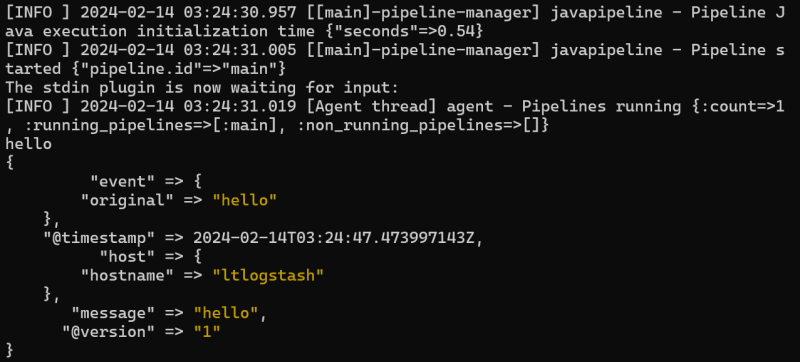
{

stdout

{

}

}

* Now run the pipeline manually sudo /usr/share/logstash/bin/logstash -f ~/basic.conf  
  
* Lets create a pipeline which reads input from stdin and writes the output to stdout and also a file /tmp/second

input

{

stdin

{

tags => ["learning", "pipeline"]

}

}

output

{

stdout

{

}

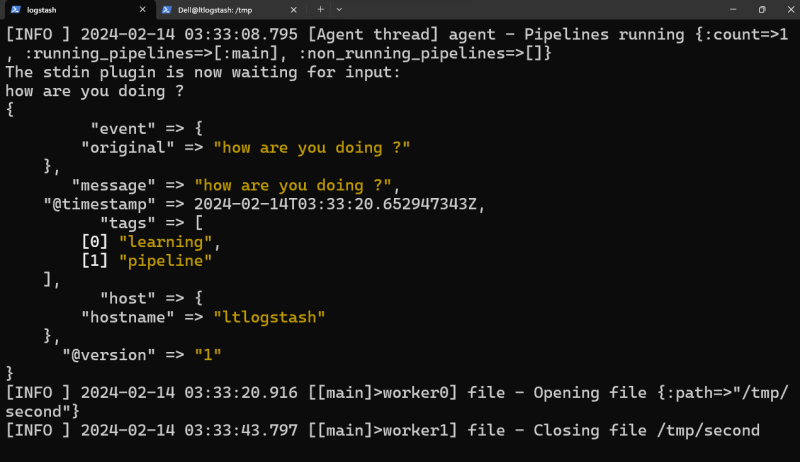
file

{

path => "/tmp/second"

}

}

* Now run the pipeline  
  
* Lets create a pipeline which reads from apache logs and writes the output to stdout

input

{

file

{

path => [ "/var/log/apache2/access.log"]

}

}

output

{

stdout {}

}

* Lets try a filter plugin

input

{

stdin {}

}

filter

{

mutate

{

split => { "hostname" => "." }

add\_field => { "shortHostname" => "%{[hostname][0]}" }

}

}

output

{

stdout {}

}

**Share this:**