MENU ☰

MAY 17, 2023

DevOps Classroomnotes 17/May/2023

Logstash

- Lets create a linux vm and explore logstash

Logstash pipeline:

- Logstash pipeline syntax

```
input {}
filter {}
output {}
```

- In input section we can define the datasources from where we process inputs `Extract`
- In Filter section we define the transformations `Transform`
- In output section we define the destination `Load`
- The list of inputs is all the installed logstash input plugins and same with other sections

Lets create a very basic pipeline which reads input from stdin and displays out to stdout

- Stdin input plugin Refer Here
- Stdout output plugin Refer Here
- Pipeline

```
input {
    stdin {
    }
}
output {
    stdout {
    }
}
```

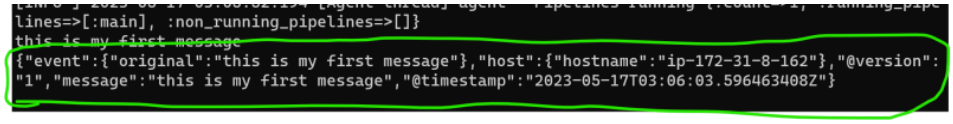
- Create a file with above content in `/tmp/first.conf`
- cd in `/usr/share/logstash` and execute the following command `sudo ./bin/logstash -f /tmp/first.conf`

```
ipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}

this is my first input
{
          "host" => {
        "hostname" => "ip-172-31-8-162"
    },
      "@version" => "1",
       "message" => "this is my first input",
    "@timestamp" => 2023-05-17T03:01:51.185072690Z,
         "event" => {
        "original" => "this is my first input"
    }
}
```

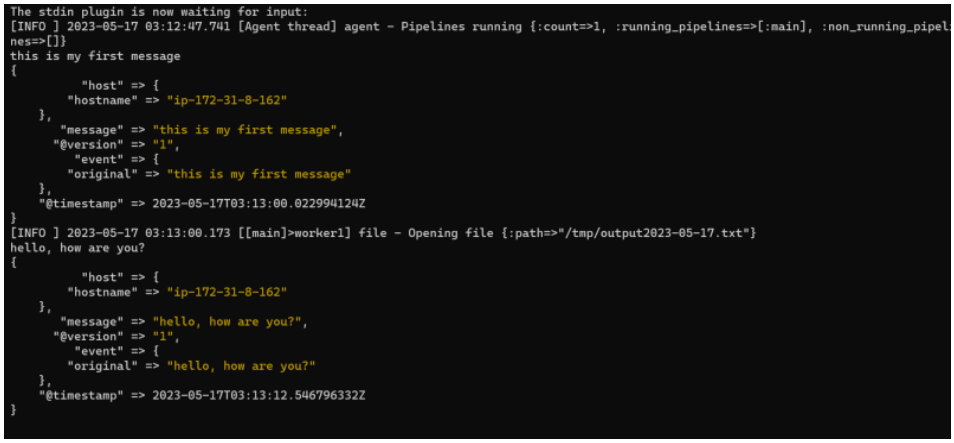
- Now lets the codec from rubydebug to json
- Edit first.conf with following content and start logstash `sudo ./bin/logstash -f /tmp/first.conf`

```
input {
    stdin {
    }
}
output {
    stdout {
        codec => json
    }
}
```

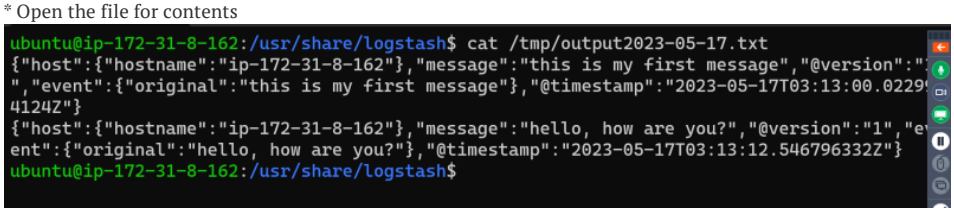


* Lets add one more output to some file `stdout => codec => rubydebug
* Refer Here for file output plugin

```
input {
    stdin {
    }
}
output {
    stdout {
    }
    file {
        path => "/tmp/output%{+YYYY-MM-dd}.txt"
    }
}
```



* Open the file for contents



Activity 2: Lets create a pipeline to read the file /tmp/test and display the contents in stdout

- input = file
- output = stdout

```
input {
    file {
        path => ["/tmp/test"]
    }
}
output {
    stdout {

    }
}
```

- install apache and redirect `/var/log/apache2/access.log` to stdout

```
input {
    file {
        path => ["/var/log/apache2/access.log"]
    }
}
output {
    stdout {

    }
}
```

- Lets try to understand filters.
- Grok filter can parse unstructured data into fields Refer Here

Leave a Reply

Enter your comment here...

About continuous learner

devops & cloud enthusiastic learner

VIEW ALL POSTS

PREVIOUS POST

Azure Classroomnotes 17/May/2023

NEXT POST

AWS Classroomnotes 17/May/2023