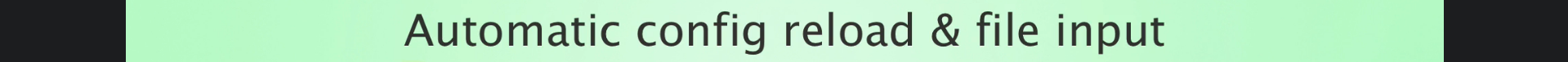
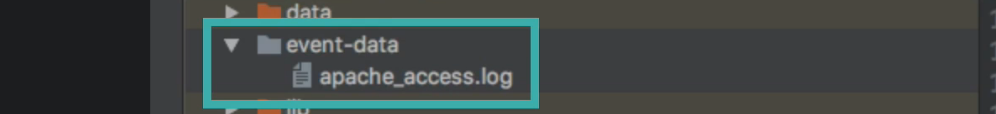
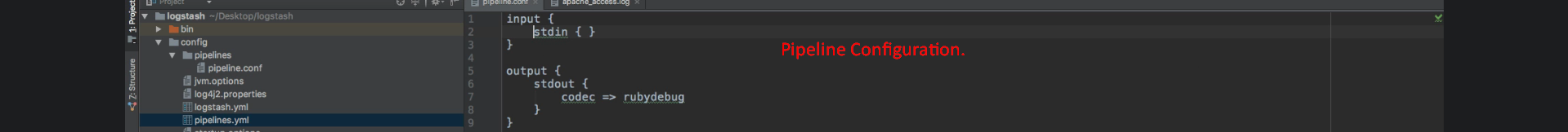
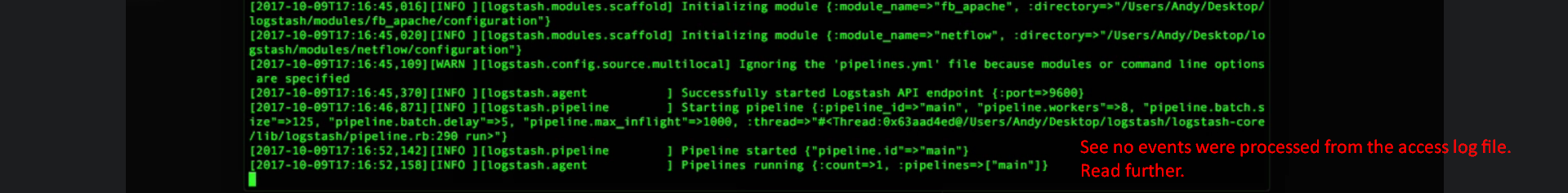
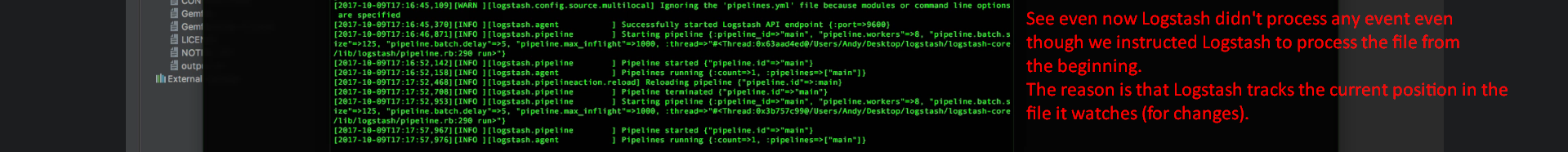
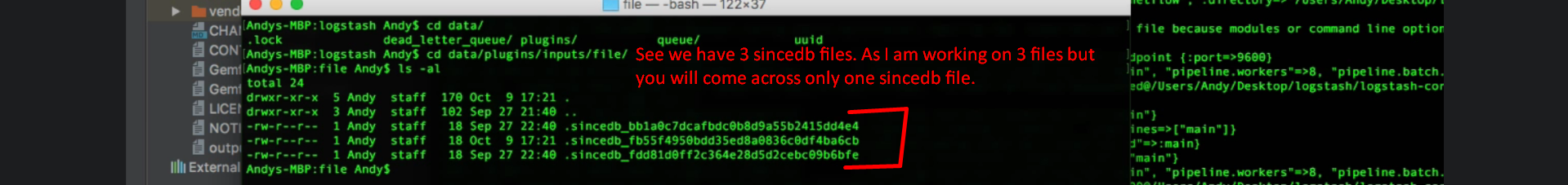
1. 
2. Let’s set up a very simple **pipeline configuration** which we can expand on in the following lectures.
3. Let us see a text file with which we will be working, and it contains HTTP Request.
   1. The format of the requests in this file is **combined format** which is more common way to format **access log**.
   2. Jatin there are two formats for **Access Log for a Web Server**:
      1. **Common Log Format**:
         1. <http://fileformats.archiveteam.org/wiki/Common_Log_Format>
      2. **Combined Log Format**:
         1. <http://fileformats.archiveteam.org/wiki/Combined_Log_Format>
4. Location for the access log file.   
     
   The file can be placed anywhere we want.
5. The following pipeline configuration file contains the configuration for listening for events from the Command Line and output the processed events back on the Command Line.  
   
6. Now we want automatic reloading of our configuration file.  
   Till now to reload, we have to restart the Logstash which was inconvenient.
7. Luckily, we have a solution for automatic configuration reloading but not without a catch.
8. To instruct Logstash to automatically reload the configuration file, we just need to pass an argument when starting Logstash.  
   **--config.reload.automatic**
9. A screen shot of a computer

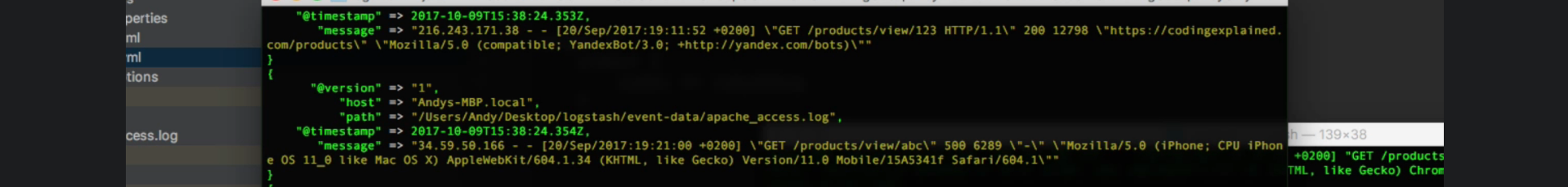
   Description automatically generated with low confidence  
   But this pipeline is not that useful.  
   Let’s change the STDIN Input Plugin to the File Input Plugin instead (to read from the text file : Access Log File).  
   A screen shot of a computer

   Description automatically generated with medium confidence  
     
   When we go back to the terminal (without doing anything), we see this error (**can’t reload pipeline, because the existing pipeline is not reloadable**) which is not directly related to the real error.  
   A screen shot of a computer

   Description automatically generated with medium confidence  
   The error was created by the STDIN Input Plugin which doesn’t support auto reloading.  
     
   Why automatic configuration reloading is not allowed for STDIN Input Plugin.  
   **Jatin**:  
   A picture containing text, screenshot, font

   Description automatically generated  
   <https://www.elastic.co/guide/en/logstash/current/reloading-config.html#:~:text=edit-,Input%20and%20output,-plugins%20usually%20interact>
10. Let’s restart the Logstash to reload the new configuration ( **we defined above having file input plugin** )  
    
11. Notice that no events have been processed from the file.
12. This is because **by default** File Input Plugin watches the file for changes from the end of the file.  
    That means it processes any lines that we add to the file moving forward (after starting the pipeline having file input plugin).  
    We can change it with option 🡺 **start\_position=”beginning”**
13. A screenshot of a computer

    Description automatically generated with medium confidence  
    See now the Logstash automatically reloading the pipe configuration file.  
    Specifically, Logstash records the **byte offset** indicating which part of a file has been already processed.  
    This is to avoid processing the same events multiple times.  
    **For example**: With configuration start\_position => “beginning”, we don’t want each time we restart Logstash, Logstash reprocess those already processed events.   
    This means that start\_position => “beginning” is useful when first time Logstash encounters a file.   
    As soon as Logstash encounters a file, it maintains byte offset for the file.
14. This Byte Offset info is kept in memory and when we shut down Logstash, the info is saved to a file.
15. Let’s stop the Logstash.
16. The file containing the progress is called **sincedb file**.
17. For each file being watched for changes, a sincedb file is maintained.
18. Let us look for this file in Data Directory.
19. 
20. Let’s see what is inside the file sincedb.
21. A screenshot of a computer

    Description automatically generated with medium confidence
22. Let’s restart the Logstash.
23. This time as we can see the Logstash has processed each line in the access log file as an events.
24. 
25. If we restart the Logstash now, it will not reprocess the lines in the file because a file sincedb has been created with Byte Offset.
26. Let’s add a new line to the file to verify that Logstash is indeed watching the file for the changes.  
      
      
    A screen shot of a computer

    Description automatically generated with low confidence  
      
    So, we’re reading the access file as events successfully.
27. As while implementing the pipeline configuration, we’re going to need to repeatedly test the configuration out with the access log.  
    For testing, we need to add lines to the access log file with the command we saw earlier.   
    Better to use easier interface we saw earlier to modify the event that Logstash should process.  
    To do this, we will add **HTTP Input Plugin** to send test events to Logstash. A screenshot of a computer

    Description automatically generated with medium confidence  
    Comment in the Pipeline configuration can be added with #.  
      
    See, the Pipeline Configuration will be auto-reloaded.  
      
    A picture containing text, screenshot, diagram, font

    Description automatically generated
28. Now we know how Logstash replaces the pipeline configuration when auto-reload configuration is enabled, and we change the configuration in a running Logstash’s configuration.
29. Let’s quickly send HTTP request to see if new configuration works.
30. 
31. A screen shot of a computer

    Description automatically generated with low confidence
32. **Recap**:
    1. How to use file input plugin.
    2. How to instruct file input plugin to process files from the beginning.
    3. sincedb file is created to track of how much of a file has already been processed in order to avoid the processing of the same event more than once.
    4. How to enable automatic configuration reloading(auto reload of changed pipeline configuration) and Logstash handles reloading internally.
    5. We added HTTP Input Plugin to make it easier to test our pipeline throughout the session of the course.