- **file_id** Identifies a single file, or the set of files to be parsed. The format of this id is:

  ```
  [container_name:]<prog>.[stdin | stdout | prgout]
  ```

  Where `prog` is a program or utility name whose stdin, stdout, or program output (prgout) artifacts will include timestamps. The optional `container_name` identifies the container hosting the file. Labs with a single container can omit this qualifier. Alternately, an explicit `file_path` is intended for log files of services that persist across multiple student operations. If the given path is not absolute, it is relative to the container user's home directory. The wildcard character '*' can be used in place of `prog`, i.e., *.stdin is for all stdin artifacts and *.stdout is for all stdout artifacts. Note prestop files are excluded from wildcard results.

- **field_type** The following `field_type`'s are used to identify fields within a selected line in the file, as determined by the `line_type` and `line_id` defined further below. Once the line is found, the `field_type` and the `field_id` locate the value within the line.

  - **TOKEN** Treat the line as space-delimited tokens
  - **PARENS** The desired value is contained in parenthesis
  - **QUOTES** The desired value is contained in quotes
  - **SLASH** The desired value is contained within slashes, e.g., /foo/
  - **SEARCH** The result is assigned the value of the search defined by the given `field_id`, which is treated as an expression having the syntax of pythons parse.search function. E.g., `frame.number=={:d}` would yield the frame number.
  - **GROUP** Intended for use with "REGEX" line types, the result is set to the value of the regex group number named by the `field_id`. Regular expressions and their groups are processed using the python re.search semantics.

- **line_type** Each of the above `field_type`'s require a `line_type` and `line_id` to locate the line within the file. The `line_type` value is one of the following:

  - **LINE** – The `line_id` is an integer line number (starting at one). Use of this to identify lines is discouraged since minor lab changes might alter the count.
  - **STARTSWITH** – the `line_id` is a string. This names the first occurrence of a line that starts with this string.
  - **HAVESTRING** – The `line_id` is a string. This names the first occurrence of a line that contains the string.
  - **REGEX** – The `line_id` is a regular expression. This names the first occurrence of a line that matches the regular expression. Also see the "GROUP" field_type.
  - **NEXT_STARTSWITH** – the `line_id` is a string. This names the line preceeding the first occurrence of a line that starts with this string.
  - **HAVESTRING_TS** – Intended for use with log files that have timestamped entries. Each entry containing the string identified in `line_id` will have its result stored as a timestamped value as if it came from a timestamped stdout or stdin file. See the snort lab for an example.
  - **REGEX_TS** – Similar to HAVESTRING_TS, but with REGEX semantics, including optional use of the GROUP `field_type`.