### parse

Parses data using a predefined parser or user-defined rule.

#### Syntax

parse [overlay=BOOL] PARSERparse [OPTIONS] PARSING\_RULE, ...

Required Parameter

**PARSER**

Predefined parser name. You can see the names of the parsers in the web console.

1. (STD, ENT) Check the **Name** in **SYSTEM CONFIG > PARSER/TRANSFORMER > Parser**
2. (MAE, SNR) Check the **Parser code** in **Logger > Parser**, **Normalizer**.

This option cannot be used with the field=TARGET\_FIELD option.

**PARSING\_RULE, ...**

User-defined rules, separated by a comma (,). The syntax of a parsing rule is "START\_ANCHOR\*STOP\_ANCHOR" as FIELD\_NAME:

1. START\_ANCHOR\*STOP\_ANCHOR: Parse anchors
2. as FIELD\_NAME: Label given as the name of the field

The parse command parses strings according to specified start and stop anchors, and then labels them with FIELD\_NAME.

Optional Parameter

**overlay=BOOL**

Option to control data-overwrite (default: f).

1. t: Outputs the parsed data and the original data.
2. f: Outputs the parsed data only.

**field=TARGET\_FIELD**

Field to be parsed in the input data stream (default: line). This option cannot be used with the PARSER argument.

#### Usage

Parse the logs stored in the ssh\_log table using the openssh parser.

table from=20200601 to=20200701 ssh\_log | parse openssh

Extract the field by specifying the start and end text from the log (using the following contents by saving it as "sample.txt").

Nov 11 00:00:00 session: Proto:17, Policy:pass, Rule:9000, Type:open, Start\_Time:Nov 11 00:00:00, End\_Time:-

You can extract the **session**, **proto**, **policy**, **rule**, and **end\_time** fields from the above source using the command below.

textfile /opt/logpresso/sample.txt | parse "session:\* " as session, "Proto:\*," as proto, "Policy:\*," as policy, "Rule:\*," as rule, "Start\_Time:\*," as start\_time, "End\_Time:\*" as end\_time