LogCLI getting started

- Install logcli
  - Binary (Recommended)
  - Build LogCLI from source
- Set up command completion
- LogCLI usage
  - Authenticate to Loki
- LogCLI command reference
  - query command reference
  - instant-query command reference
  - labels command reference
  - series command reference
  - stats command reference
  - volume command reference
  - volume_range command reference
  - detected-fields command reference
  - delete command reference
  - delete create command reference
  - delete list command reference
  - delete cancel command reference
  - Use --stdin to query locally
    - Notes on stdin usage
    - stdin examples
- Batching
- logcli example queries
- Troubleshoot logcli

logcli is a command-line client for Loki that lets you run LogQL queries against your Loki instance. The `query` command will output extra information about the query and its results, such as the API URL, set of common labels, and set of excluded labels.

This is useful, for example, if you want to download a range of logs from Loki. Or want to perform analytical administration tasks, for example, discover the number of log streams to understand your label cardinality, or find out the estimated volume of data that a query will search over. You can also use logcli as part of shell scripts.

If you are a Grafana Cloud user, you can also use logcli to query logs that you have exported to long-term storage with Cloud Logs Export, or any other Loki formatted log data.

{{< admonition type="note" >}} Note that logcli is a querying tool, it cannot be used to ingest logs. {{< /admonition >}}

Install logcli

As a best practice, you should download the version of logcli that matches your Loki version. And upgrade your logcli when you upgrade your version of Loki.

Binary (Recommended)

Download the `logcli` binary from the Loki releases page.

Builds are available for Linux, Mac, and Windows.

Build LogCLI from source

Clone the Loki repository and build `logcli` from source:

```
git clone https://github.com/grafana/loki.git
cd loki
make logcli
```

Optionally, move the binary into a directory that is part of your `$PATH`.

```
cp cmd/logcli/logcli /usr/local/bin/logcli
```

Set up command completion

You can set up tab-completion for `logcli` with one of the two options, depending on your shell:

- For bash, add this to your `~/.bashrc` file:

```
eval "$(logcli --completion-script-bash)"
```

- For zsh, add this to your `~/.zshrc` file:

```
eval "$(logcli --completion-script-zsh)"
```

LogCLI usage

Once you have installed logcli, you can run it in the following way:

`logcli <command> [<flags>, <args> ...]`

`<command>` points to one of the commands, detailed in the command reference below.

`<flags>` is one of the subcommands available for each command.

`<args>` is a list of space separated arguments. Arguments can optionally be overridden using environment variables. Environment variables will always take precedence over command line arguments.

Authenticate to Loki

To connect to a Loki instance, set the following argument:

- `--addr=http://loki.example.com:3100` or the `LOKI_ADDR` environment variable

For example, to query a local Loki instance directly without needing a username and password:

```
export LOKI_ADDR=http://localhost:3100


logcli query '{service_name="website"}'
```

To connect to a Loki instance which requires authentication, you will need to additionally set the following arguments:

- `--username` or the `LOKI_USERNAME` environment variable
- `--password` or the `LOKI_PASSWORD` environment variable

For example, to query Grafana Cloud:

```
export LOKI_ADDR=https://logs-us-west1.grafana.net
export LOKI_USERNAME=<username>
export LOKI_PASSWORD=<password>

logcli query '{service_name="website"}'
```

To specify a particular tenant, set the following argument:

- `--org-id` or the `LOKI_ORG_ID` environment variable

{{< admonition type="note" >}} If you are running Loki behind a proxy server and you have authentication configured, you will also have to pass in LOKI_USERNAME and LOKI_PASSWORD, LOKI_BEARER_TOKEN or LOKI_BEARER_TOKEN_FILE accordingly. {{< /admonition >}}

LogCLI command reference

The output of `logcli help`:

```
usage: logcli [<flags>] <command> [<args> ...]

A command-line for loki.


Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
  -o, --output=default        Specify output mode [default, raw, jsonl].
                              raw suppresses log labels and timestamp.
  -z, --timezone=Local        Specify the timezone to use when formatting output
                              timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                              Specify the format of timestamps in the default
                              output mode [rfc3339, rfc3339nano, rfc822z,
                              rfc1123z, stampmicro, stampmilli, stampnano,
                              unixdate]
      --cpuprofile=""         Specify the location for writing a CPU profile.
```

```
--memprofile=""         Specify the location for writing a memory profile.
--[no-]stdin            Take input logs from stdin
--addr="http://localhost:3100"
                        Server address. Can also be set using LOKI_ADDR
                        env var. ($LOKI_ADDR)
--username=""           Username for HTTP basic auth. Can also be set
                        using LOKI_USERNAME env var. ($LOKI_USERNAME)
--password=""           Password for HTTP basic auth. Can also be set
                        using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
--ca-cert=""            Path to the server Certificate Authority.
                        Can also be set using LOKI_CA_CERT_PATH env var.
                        ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                        be set using LOKI_TLS_SKIP_VERIFY env var.
                        ($LOKI_TLS_SKIP_VERIFY)
--cert=""               Path to the client certificate. Can also
                        be set using LOKI_CLIENT_CERT_PATH env var.
                        ($LOKI_CLIENT_CERT_PATH)
--key=""                Path to the client certificate key. Can also
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
--org-id=""             adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
--query-tags=""         adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
--[no-]nocache          adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
--bearer-token=""       adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
--bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
```

```
                            ($LOKI_BEARER_TOKEN_FILE)
      --retries=0           How many times to retry each query when
                            getting an error response from Loki. Can also
                            be set using LOKI_CLIENT_RETRIES env var.
                            ($LOKI_CLIENT_RETRIES)
      --min-backoff=0       Minimum backoff time between retries. Can also
                            be set using LOKI_CLIENT_MIN_BACKOFF env var.
                            ($LOKI_CLIENT_MIN_BACKOFF)
      --max-backoff=0       Maximum backoff time between retries. Can also
                            be set using LOKI_CLIENT_MAX_BACKOFF env var.
                            ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                            The authorization header used. Can also
                            be set using LOKI_AUTH_HEADER env var.
                            ($LOKI_AUTH_HEADER)
      --proxy-url=""        The http or https proxy to use when
                            making requests. Can also be set
                            using LOKI_HTTP_PROXY_URL env var.
                            ($LOKI_HTTP_PROXY_URL)
      --[no-]compress       Request that Loki compress returned
                            data in transit. Can also be set
                            using LOKI_HTTP_COMPRESSION env var.
                            ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy       Use ProxyFromEnvironment to use net/http
                            ProxyFromEnvironment configuration, eg HTTP_PROXY
                            ($LOKI_ENV_PROXY)
      --log.level           Only log messages with the given severity or above. Valid
levels: [debug, info, warn, error]

Commands:
help [<command>...]
    Show help.

query [<flags>] <query>
    Run a LogQL query.

    The "query" command is useful for querying for logs. Logs can be returned in
    a few output modes:

      raw: log line
```

```
  default: log timestamp + log labels + log line
  jsonl: JSON response from Loki API of log line

The output of the log can be specified with the "-o" flag, for example,
"-o raw" for the raw output format.

The "query" command will output extra information about the query and its
results, such as the API URL, set of common labels, and set of excluded
labels. This extra information can be suppressed with the --quiet flag.

By default we look over the last hour of data; use --since to modify or
provide specific start and end times with --from and --to respectively.

Notice that when using --from and --to then ensure to use RFC3339Nano time
format, but without timezone at the end. The local timezone will be added
automatically or if using --timezone flag. In default output mode the
--output-timestamp-format flag can be used to modify the output timestamp.

Example:

  logcli query
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output=jsonl
     'my-query'

Example with --output-timestamp-format:

  logcli query
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output-timestamp-format=rfc3339nano
     'my-query'

The output is limited to 30 entries by default; use --limit to increase.

While "query" does support metrics queries, its output contains multiple
data points between the start and end query time. This output is used to
```

```
build graphs, similar to what is seen in the Grafana Explore graph view.
If you are querying metrics and just want the most recent data point (like
what is seen in the Grafana Explore table view), then you should use the
"instant-query" command instead.

Parallelization:

You can download an unlimited number of logs in parallel, there are a few
flags which control this behaviour:

  --parallel-duration
  --parallel-max-workers
  --part-path-prefix
  --overwrite-completed-parts
  --merge-parts
  --keep-parts

Refer to the help for each flag for details about what each of them do.

Example:

  logcli query
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output=jsonl
     --parallel-duration="15m"
     --parallel-max-workers="4"
     --part-path-prefix="/tmp/my_query"
     --merge-parts
     'my-query'

This example will create a queue of jobs to execute, each being 15 minutes
in duration. In this case, that means, for the 10-hour total duration,
there will be forty 15-minute jobs. The --limit flag is ignored.

It will start four workers, and they will each take a job to work on from
the queue until all the jobs have been completed.

Each job will save a "part" file to the location specified by the
```

```
    --part-path-prefix. Different prefixes can be used to run multiple queries
    at the same time. The timestamp of the start and end of the part is in
    the file name. While the part is being downloaded, the filename will
    end in ".part", when it is complete, the file will be renamed to remove
    this ".part" extension. By default, if a completed part file is found,
    that part will not be downloaded again. This can be overridden with the
    --overwrite-completed-parts flag.

    Part file example using the previous command, adding --keep-parts so they
    are not deleted:

    Since we don't have the --forward flag, the parts will be downloaded in
    reverse. Two of the workers have finished their jobs (last two files),
    and have picked up the next jobs in the queue. Running ls, this is what we
    should expect to see.

    $ ls -1 /tmp/my_query*
    /tmp/my_query_20210119T183000_20210119T184500.part.tmp
    /tmp/my_query_20210119T184500_20210119T190000.part.tmp
    /tmp/my_query_20210119T190000_20210119T191500.part.tmp
    /tmp/my_query_20210119T191500_20210119T193000.part.tmp
    /tmp/my_query_20210119T193000_20210119T194500.part
    /tmp/my_query_20210119T194500_20210119T200000.part

    If you do not specify the --merge-parts flag, the part files will be
    downloaded, and logcli will exit, and you can process the files as you wish.
    With the flag specified, the part files will be read in order, and the
    output printed to the terminal. The lines will be printed as soon as the
    next part is complete, you don't have to wait for all the parts to download
    before getting output. The --merge-parts flag will remove the part files
    when it is done reading each of them. To change this, you can use the
    --keep-parts flag, and the part files will not be removed.

instant-query [<flags>] <query>
    Run an instant LogQL query.

    The "instant-query" command is useful for evaluating a metric query for a
    single point in time. This is equivalent to the Grafana Explore table view;
    if you want a metrics query that is used to build a Grafana graph, you
    should use the "query" command instead.
```

```
    This command does not produce useful output when querying for log lines;
    you should always use the "query" command when you are running log queries.

    For more information about log queries and metric queries, refer to the
    LogQL documentation:

    https://grafana.com/docs/loki/latest/logql/

labels [<flags>] [<label>]
    Find values for a given label.

series [<flags>] <matcher>
    Run series query.

    The "series" command will take the provided label matcher and return all the
    log streams found in the time window.

    It is possible to send an empty label matcher '{}' to return all streams.

    Use the --analyze-labels flag to get a summary of the labels found in all
    streams. This is helpful to find high cardinality labels.

fmt
    Formats a LogQL query.

stats [<flags>] <query>
    Run a stats query.

    The "stats" command will take the provided query and return statistics
    from the index on how much data is contained in the matching stream(s).
    This only works against Loki instances using the TSDB index format.

    By default we look over the last hour of data; use --since to modify or
    provide specific start and end times with --from and --to respectively.

    Notice that when using --from and --to then ensure to use RFC3339Nano time
    format, but without timezone at the end. The local timezone will be added
    automatically or if using --timezone flag.
```

```
    Example:

      logcli stats
         --timezone=UTC
         --from="2021-01-19T10:00:00Z"
         --to="2021-01-19T20:00:00Z"
         'my-query'

volume [<flags>] <query>
    Run a volume query.

    The "volume" command will take the provided label selector(s) and return
    aggregate volumes for series matching those volumes. This only works against
    Loki instances using the TSDB index format.

    By default we look over the last hour of data; use --since to modify or
    provide specific start and end times with --from and --to respectively.

    Notice that when using --from and --to then ensure to use RFC3339Nano time
    format, but without timezone at the end. The local timezone will be added
    automatically or if using --timezone flag.

    Example:

      logcli volume
         --timezone=UTC
         --from="2021-01-19T10:00:00Z"
         --to="2021-01-19T20:00:00Z"
         'my-query'

volume_range [<flags>] <query>
    Run a volume query and return timeseries data.

    The "volume_range" command will take the provided label selector(s) and
    return aggregate volumes for series matching those volumes, aggregated into
    buckets according to the step value. This only works against Loki instances
    using the TSDB index format.

    By default we look over the last hour of data; use --since to modify or
    provide specific start and end times with --from and --to respectively.
```

```
    Notice that when using --from and --to then ensure to use RFC3339Nano time
    format, but without timezone at the end. The local timezone will be added
    automatically or if using --timezone flag.

    Example:

      logcli volume_range
        --timezone=UTC
        --from="2021-01-19T10:00:00Z"
        --to="2021-01-19T20:00:00Z"
         --step=1h
        'my-query'

detected-fields [<flags>] <query> [<field>]
    Run a query for detected fields..

delete <command> [<args> ...]
    Manage log deletion requests.

    The "detected-fields" command will return information about fields detected
    using either the "logfmt" or "json" parser against the log lines returned by
    the provided query for the provided time range.

    The "detected-fields" command will output extra information about the query
    and its results, such as the API URL, set of common labels, and set of
    excluded labels. This extra information can be suppressed with the --quiet
    flag.

    By default we look over the last hour of data; use --since to modify or
    provide specific start and end times with --from and --to respectively.

    Notice that when using --from and --to then ensure to use RFC3339Nano time
    format, but without timezone at the end. The local timezone will be added
    automatically or if using --timezone flag.

    Example:

      logcli detected-fields
         --timezone=UTC
```

```
        --from="2021-01-19T10:00:00Z"
        --to="2021-01-19T20:00:00Z"
        --output=jsonl
        'my-query'

    The output is limited to 100 fields by default; use --field-limit to
    increase. The query is limited to processing 1000 lines per subquery;
    use --line-limit to increase.
```

`query` command reference

The output of `logcli help query`:

```
usage: logcli query [<flags>] <query>

Run a LogQL query.

The "query" command is useful for querying for logs. Logs can be returned in a
few output modes:

  raw: log line
  default: log timestamp + log labels + log line
  jsonl: JSON response from Loki API of log line

The output of the log can be specified with the "-o" flag, for example, "-o raw"
for the raw output format.

The "query" command will output extra information about the query and its
results, such as the API URL, set of common labels, and set of excluded labels.
This extra information can be suppressed with the --quiet flag.

By default we look over the last hour of data; use --since to modify or provide
specific start and end times with --from and --to respectively.

Notice that when using --from and --to then ensure to use RFC3339Nano time
format, but without timezone at the end. The local timezone will be added
automatically or if using --timezone flag. In default output mode the
--output-timestamp-format flag can be used to modify the output timestamp.
```

```
Example:

  logcli query
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output=jsonl
     'my-query'

Example with --output-timestamp-format:

  logcli query
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output-timestamp-format=rfc3339nano
     'my-query'

The output is limited to 30 entries by default; use --limit to increase.

While "query" does support metrics queries, its output contains multiple data
points between the start and end query time. This output is used to build
graphs, similar to what is seen in the Grafana Explore graph view. If you are
querying metrics and just want the most recent data point (like what is seen in
the Grafana Explore table view), then you should use the "instant-query" command
instead.

Parallelization:

You can download an unlimited number of logs in parallel, there are a few flags
which control this behaviour:

  --parallel-duration
  --parallel-max-workers
  --part-path-prefix
  --overwrite-completed-parts
  --merge-parts
  --keep-parts

Refer to the help for each flag for details about what each of them do.
```

```
Example:

  logcli query
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output=jsonl
     --parallel-duration="15m"
     --parallel-max-workers="4"
     --part-path-prefix="/tmp/my_query"
     --merge-parts
     'my-query'

This example will create a queue of jobs to execute, each being 15 minutes in
duration. In this case, that means, for the 10-hour total duration, there will
be forty 15-minute jobs. The --limit flag is ignored.

It will start four workers, and they will each take a job to work on from the
queue until all the jobs have been completed.

Each job will save a "part" file to the location specified by the
--part-path-prefix. Different prefixes can be used to run multiple queries at
the same time. The timestamp of the start and end of the part is in the file
name. While the part is being downloaded, the filename will end in ".part",
when it is complete, the file will be renamed to remove this ".part" extension.
By default, if a completed part file is found, that part will not be downloaded
again. This can be overridden with the --overwrite-completed-parts flag.

Part file example using the previous command, adding --keep-parts so they are
not deleted:

Since we don't have the --forward flag, the parts will be downloaded in reverse.
Two of the workers have finished their jobs (last two files), and have picked up
the next jobs in the queue. Running ls, this is what we should expect to see.

$ ls -1 /tmp/my_query* /tmp/my_query_20210119T183000_20210119T184500.part.tmp
/tmp/my_query_20210119T184500_20210119T190000.part.tmp
/tmp/my_query_20210119T190000_20210119T191500.part.tmp
/tmp/my_query_20210119T191500_20210119T193000.part.tmp
```

```
/tmp/my_query_20210119T193000_20210119T194500.part
/tmp/my_query_20210119T194500_20210119T200000.part

If you do not specify the --merge-parts flag, the part files will be downloaded,
and logcli will exit, and you can process the files as you wish. With the flag
specified, the part files will be read in order, and the output printed to
the terminal. The lines will be printed as soon as the next part is complete,
you don't have to wait for all the parts to download before getting output.
The --merge-parts flag will remove the part files when it is done reading each
of them. To change this, you can use the --keep-parts flag, and the part files
will not be removed.


Flags:
      --[no-]help               Show context-sensitive help (also try
                                --help-long and --help-man).
      --[no-]version            Show application version.
  -q, --[no-]quiet              Suppress query metadata
      --[no-]stats              Show query statistics
  -o, --output=default          Specify output mode [default, raw, jsonl].
                                raw suppresses log labels and timestamp.
  -z, --timezone=Local          Specify the timezone to use when formatting
                                output timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                                Specify the format of timestamps in the default
                                output mode [rfc3339, rfc3339nano, rfc822z,
                                rfc1123z, stampmicro, stampmilli, stampnano,
                                unixdate]
      --cpuprofile=""           Specify the location for writing a CPU profile.
      --memprofile=""           Specify the location for writing a memory
                                profile.
      --[no-]stdin              Take input logs from stdin
      --addr="http://localhost:3100"
                                Server address. Can also be set using LOKI_ADDR
                                env var. ($LOKI_ADDR)
      --username=""             Username for HTTP basic auth. Can also be set
                                using LOKI_USERNAME env var. ($LOKI_USERNAME)
      --password=""             Password for HTTP basic auth. Can also be set
                                using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
      --ca-cert=""              Path to the server Certificate Authority.
```

```
                          Can also be set using LOKI_CA_CERT_PATH env var.
                          ($LOKI_CA_CERT_PATH)
    --[no-]tls-skip-verify    Server certificate TLS skip verify. Can also
                          be set using LOKI_TLS_SKIP_VERIFY env var.
                          ($LOKI_TLS_SKIP_VERIFY)
    --cert=""             Path to the client certificate. Can also
                          be set using LOKI_CLIENT_CERT_PATH env var.
                          ($LOKI_CLIENT_CERT_PATH)
    --key=""              Path to the client certificate key. Can also
                          be set using LOKI_CLIENT_KEY_PATH env var.
                          ($LOKI_CLIENT_KEY_PATH)
    --org-id=""           adds X-Scope-OrgID to API requests for
                          representing tenant ID. Useful for requesting
                          tenant data when bypassing an auth gateway.
                          Can also be set using LOKI_ORG_ID env var.
                          ($LOKI_ORG_ID)
    --query-tags=""       adds X-Query-Tags http header to API requests.
                          This header value will be part of `metrics.go`
                          statistics. Useful for tracking the query.
                          Can also be set using LOKI_QUERY_TAGS env var.
                          ($LOKI_QUERY_TAGS)
    --[no-]nocache        adds Cache-Control: no-cache http header to API
                          requests. Can also be set using LOKI_NO_CACHE
                          env var. ($LOKI_NO_CACHE)
    --bearer-token=""     adds the Authorization header to API
                          requests for authentication purposes.
                          Can also be set using LOKI_BEARER_TOKEN env var.
                          ($LOKI_BEARER_TOKEN)
    --bearer-token-file=""    adds the Authorization header to API requests
                          for authentication purposes. Can also be
                          set using LOKI_BEARER_TOKEN_FILE env var.
                          ($LOKI_BEARER_TOKEN_FILE)
    --retries=0           How many times to retry each query when
                          getting an error response from Loki. Can also
                          be set using LOKI_CLIENT_RETRIES env var.
                          ($LOKI_CLIENT_RETRIES)
    --min-backoff=0       Minimum backoff time between retries. Can also
                          be set using LOKI_CLIENT_MIN_BACKOFF env var.
                          ($LOKI_CLIENT_MIN_BACKOFF)
    --max-backoff=0       Maximum backoff time between retries. Can also
```

```
                          be set using LOKI_CLIENT_MAX_BACKOFF env var.
                          ($LOKI_CLIENT_MAX_BACKOFF)
    --auth-header="Authorization"
                          The authorization header used. Can also
                          be set using LOKI_AUTH_HEADER env var.
                          ($LOKI_AUTH_HEADER)
    --proxy-url=""        The http or https proxy to use when
                          making requests. Can also be set
                          using LOKI_HTTP_PROXY_URL env var.
                          ($LOKI_HTTP_PROXY_URL)
    --[no-]compress       Request that Loki compress returned
                          data in transit. Can also be set
                          using LOKI_HTTP_COMPRESSION env var.
                          ($LOKI_HTTP_COMPRESSION)
    --[no-]envproxy       Use ProxyFromEnvironment to use net/http
                          ProxyFromEnvironment configuration, eg
                          HTTP_PROXY ($LOKI_ENV_PROXY)
    --limit=30            Limit on number of entries to print. Setting it
                          to 0 will fetch all entries.
    --since=1h            Lookback window.
    --from=FROM           Start looking for logs at this absolute time
                          (inclusive)
    --to=TO               Stop looking for logs at this absolute time
                          (exclusive)
    --step=STEP           Query resolution step width, for metric queries.
                          Evaluate the query at the specified step over
                          the time range.
    --interval=INTERVAL   Query interval, for log queries. Return entries
                          at the specified interval, ignoring those
                          between. **This parameter is experimental,
                          please see Issue 1779**
    --batch=1000          Query batch size to use until 'limit' is reached
    --parallel-duration=1h  Split the range into jobs of this length to
                          download the logs in parallel. This will
                          result in the logs being out of order.
                          Use --part-path-prefix to create a file per job
                          to maintain ordering.
    --parallel-max-workers=1  Max number of workers to start up for parallel
                          jobs. A value of 1 will not create any parallel
                          workers. When using parallel workers, limit is
```

```
                              ignored.
    --part-path-prefix=PART-PATH-PREFIX
                              When set, each server response will be saved
                              to a file with this prefix. Creates files in
                              the format: 'prefix-utc_start-utc_end.part'.
                              Intended to be used with the parallel-* flags
                              so that you can combine the files to maintain
                              ordering based on the filename. Default is to
                              write to stdout.
    --[no-]overwrite-completed-parts
                              Overwrites completed part files. This will
                              download the range again, and replace the
                              original completed part file. Default will skip
                              a range if it's part file is already downloaded.
    --[no-]merge-parts        Reads the part files in order and writes the
                              output to stdout. Original part files will be
                              deleted with this option.
    --[no-]keep-parts         Overrides the default behaviour of --merge-parts
                              which will delete the part files once all the
                              files have been read. This option will keep the
                              part files.
    --[no-]forward            Scan forwards through logs.
    --[no-]no-labels          Do not print any labels
    --exclude-label=EXCLUDE-LABEL ...
                              Exclude labels given the provided key during
                              output.
    --include-label=INCLUDE-LABEL ...
                              Include labels given the provided key during
                              output.
    --[no-]include-common-labels
                              Include common labels in output for each log
                              line.
    --labels-length=0         Set a fixed padding to labels
    --store-config=""         Execute the current query using a configured
                              storage from a given Loki configuration file.
    --[no-]remote-schema      Execute the current query using a remote schema
                              retrieved from the configured -schema-store.
    --schema-store=""         Store used for retrieving remote schema.
    --[no-]colored-output     Show output with colored labels
-t, --[no-]tail               Tail the logs
```

```
  -f, --[no-]follow             Alias for --tail
      --delay-for=0             Delay in tailing by number of seconds to
                                accumulate logs for re-ordering

Args:
  <query>  eg '{foo="bar",baz=~".*blip"} |~ ".*error.*"'
```

`instant-query` command reference

The output of `logcli help instant-query`:

```
usage: logcli instant-query [<flags>] <query>

Run an instant LogQL query.

The "instant-query" command is useful for evaluating a metric query for a
single point in time. This is equivalent to the Grafana Explore table view;
if you want a metrics query that is used to build a Grafana graph, you should
use the "query" command instead.

This command does not produce useful output when querying for log lines;
you should always use the "query" command when you are running log queries.

For more information about log queries and metric queries, refer to the LogQL
documentation:

https://grafana.com/docs/loki/latest/logql/



Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
  -o, --output=default        Specify output mode [default, raw, jsonl].
                              raw suppresses log labels and timestamp.
  -z, --timezone=Local        Specify the timezone to use when formatting output
                              timestamps [Local, UTC]
```

```
--output-timestamp-format=rfc3339
                        Specify the format of timestamps in the default
                        output mode [rfc3339, rfc3339nano, rfc822z,
                        rfc1123z, stampmicro, stampmilli, stampnano,
                        unixdate]
--cpuprofile=""         Specify the location for writing a CPU profile.
--memprofile=""         Specify the location for writing a memory profile.
--[no-]stdin            Take input logs from stdin
--addr="http://localhost:3100"
                        Server address. Can also be set using LOKI_ADDR
                        env var. ($LOKI_ADDR)
--username=""           Username for HTTP basic auth. Can also be set
                        using LOKI_USERNAME env var. ($LOKI_USERNAME)
--password=""           Password for HTTP basic auth. Can also be set
                        using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
--ca-cert=""            Path to the server Certificate Authority.
                        Can also be set using LOKI_CA_CERT_PATH env var.
                        ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                        be set using LOKI_TLS_SKIP_VERIFY env var.
                        ($LOKI_TLS_SKIP_VERIFY)
--cert=""               Path to the client certificate. Can also
                        be set using LOKI_CLIENT_CERT_PATH env var.
                        ($LOKI_CLIENT_CERT_PATH)
--key=""                Path to the client certificate key. Can also
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
--org-id=""             adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
--query-tags=""         adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
--[no-]nocache          adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
```

```
--bearer-token=""       adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
--bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
                        ($LOKI_BEARER_TOKEN_FILE)
--retries=0             How many times to retry each query when
                        getting an error response from Loki. Can also
                        be set using LOKI_CLIENT_RETRIES env var.
                        ($LOKI_CLIENT_RETRIES)
--min-backoff=0         Minimum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MIN_BACKOFF env var.
                        ($LOKI_CLIENT_MIN_BACKOFF)
--max-backoff=0         Maximum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MAX_BACKOFF env var.
                        ($LOKI_CLIENT_MAX_BACKOFF)
--auth-header="Authorization"
                        The authorization header used. Can also
                        be set using LOKI_AUTH_HEADER env var.
                        ($LOKI_AUTH_HEADER)
--proxy-url=""          The http or https proxy to use when
                        making requests. Can also be set
                        using LOKI_HTTP_PROXY_URL env var.
                        ($LOKI_HTTP_PROXY_URL)
--[no-]compress         Request that Loki compress returned
                        data in transit. Can also be set
                        using LOKI_HTTP_COMPRESSION env var.
                        ($LOKI_HTTP_COMPRESSION)
--[no-]envproxy         Use ProxyFromEnvironment to use net/http
                        ProxyFromEnvironment configuration, eg HTTP_PROXY
                        ($LOKI_ENV_PROXY)
--limit=30              Limit on number of entries to print. Setting it to
                        0 will fetch all entries.
--now=NOW               Time at which to execute the instant query.
--[no-]forward          Scan forwards through logs.
--[no-]no-labels        Do not print any labels
--exclude-label=EXCLUDE-LABEL ...
                        Exclude labels given the provided key during
                        output.
```

```
      --include-label=INCLUDE-LABEL ...
                               Include labels given the provided key during
                               output.
      --[no-]include-common-labels
                               Include common labels in output for each log line.
      --labels-length=0        Set a fixed padding to labels
      --store-config=""        Execute the current query using a configured
                               storage from a given Loki configuration file.
      --[no-]remote-schema     Execute the current query using a remote schema
                               retrieved from the configured -schema-store.
      --schema-store=""        Store used for retrieving remote schema.
      --[no-]colored-output    Show output with colored labels

Args:
  <query>  eg 'rate({foo="bar"} |~ ".*error.*" [5m])'
```

`labels` command reference

The output of `logcli help labels`:

```
usage: logcli labels [<flags>] [<label>]

Find values for a given label.


Flags:
      --[no-]help              Show context-sensitive help (also try --help-long
                               and --help-man).
      --[no-]version           Show application version.
  -q, --[no-]quiet             Suppress query metadata
      --[no-]stats             Show query statistics
  -o, --output=default         Specify output mode [default, raw, jsonl].
                               raw suppresses log labels and timestamp.
  -z, --timezone=Local         Specify the timezone to use when formatting output
                               timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                               Specify the format of timestamps in the default
                               output mode [rfc3339, rfc3339nano, rfc822z,
                               rfc1123z, stampmicro, stampmilli, stampnano,
```

```
                        unixdate]
--cpuprofile=""         Specify the location for writing a CPU profile.
--memprofile=""         Specify the location for writing a memory profile.
--[no-]stdin            Take input logs from stdin
--addr="http://localhost:3100"
                        Server address. Can also be set using LOKI_ADDR
                        env var. ($LOKI_ADDR)
--username=""           Username for HTTP basic auth. Can also be set
                        using LOKI_USERNAME env var. ($LOKI_USERNAME)
--password=""           Password for HTTP basic auth. Can also be set
                        using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
--ca-cert=""            Path to the server Certificate Authority.
                        Can also be set using LOKI_CA_CERT_PATH env var.
                        ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                        be set using LOKI_TLS_SKIP_VERIFY env var.
                        ($LOKI_TLS_SKIP_VERIFY)
--cert=""               Path to the client certificate. Can also
                        be set using LOKI_CLIENT_CERT_PATH env var.
                        ($LOKI_CLIENT_CERT_PATH)
--key=""                Path to the client certificate key. Can also
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
--org-id=""             adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
--query-tags=""         adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
--[no-]nocache          adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
--bearer-token=""       adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
--bearer-token-file=""  adds the Authorization header to API requests
```

```
                           for authentication purposes. Can also be
                           set using LOKI_BEARER_TOKEN_FILE env var.
                           ($LOKI_BEARER_TOKEN_FILE)
      --retries=0          How many times to retry each query when
                           getting an error response from Loki. Can also
                           be set using LOKI_CLIENT_RETRIES env var.
                           ($LOKI_CLIENT_RETRIES)
      --min-backoff=0      Minimum backoff time between retries. Can also
                           be set using LOKI_CLIENT_MIN_BACKOFF env var.
                           ($LOKI_CLIENT_MIN_BACKOFF)
      --max-backoff=0      Maximum backoff time between retries. Can also
                           be set using LOKI_CLIENT_MAX_BACKOFF env var.
                           ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                           The authorization header used. Can also
                           be set using LOKI_AUTH_HEADER env var.
                           ($LOKI_AUTH_HEADER)
      --proxy-url=""       The http or https proxy to use when
                           making requests. Can also be set
                           using LOKI_HTTP_PROXY_URL env var.
                           ($LOKI_HTTP_PROXY_URL)
      --[no-]compress      Request that Loki compress returned
                           data in transit. Can also be set
                           using LOKI_HTTP_COMPRESSION env var.
                           ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy      Use ProxyFromEnvironment to use net/http
                           ProxyFromEnvironment configuration, eg HTTP_PROXY
                           ($LOKI_ENV_PROXY)
      --since=1h           Lookback window.
      --from=FROM          Start looking for labels at this absolute time
                           (inclusive)
      --to=TO              Stop looking for labels at this absolute time
                           (exclusive)

Args:
  [<label>]  The name of the label.
```

`series` command reference

The output of `logcli help series`:

```
usage: logcli series [<flags>] <matcher>

Run series query.

The "series" command will take the provided label matcher and return all the log
streams found in the time window.

It is possible to send an empty label matcher '{}' to return all streams.

Use the --analyze-labels flag to get a summary of the labels found in all
streams. This is helpful to find high cardinality labels.


Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
  -o, --output=default        Specify output mode [default, raw, jsonl].
                              raw suppresses log labels and timestamp.
  -z, --timezone=Local        Specify the timezone to use when formatting output
                              timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                              Specify the format of timestamps in the default
                              output mode [rfc3339, rfc3339nano, rfc822z,
                              rfc1123z, stampmicro, stampmilli, stampnano,
                              unixdate]
      --cpuprofile=""         Specify the location for writing a CPU profile.
      --memprofile=""         Specify the location for writing a memory profile.
      --[no-]stdin            Take input logs from stdin
      --addr="http://localhost:3100"
                              Server address. Can also be set using LOKI_ADDR
                              env var. ($LOKI_ADDR)
      --username=""           Username for HTTP basic auth. Can also be set
                              using LOKI_USERNAME env var. ($LOKI_USERNAME)
      --password=""           Password for HTTP basic auth. Can also be set
                              using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
```

```
--ca-cert=""            Path to the server Certificate Authority.
                        Can also be set using LOKI_CA_CERT_PATH env var.
                        ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                        be set using LOKI_TLS_SKIP_VERIFY env var.
                        ($LOKI_TLS_SKIP_VERIFY)
--cert=""               Path to the client certificate. Can also
                        be set using LOKI_CLIENT_CERT_PATH env var.
                        ($LOKI_CLIENT_CERT_PATH)
--key=""                Path to the client certificate key. Can also
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
--org-id=""             adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
--query-tags=""         adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
--[no-]nocache          adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
--bearer-token=""       adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
--bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
                        ($LOKI_BEARER_TOKEN_FILE)
--retries=0             How many times to retry each query when
                        getting an error response from Loki. Can also
                        be set using LOKI_CLIENT_RETRIES env var.
                        ($LOKI_CLIENT_RETRIES)
--min-backoff=0         Minimum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MIN_BACKOFF env var.
                        ($LOKI_CLIENT_MIN_BACKOFF)
--max-backoff=0         Maximum backoff time between retries. Can also
```

```
                           be set using LOKI_CLIENT_MAX_BACKOFF env var.
                           ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                           The authorization header used. Can also
                           be set using LOKI_AUTH_HEADER env var.
                           ($LOKI_AUTH_HEADER)
      --proxy-url=""       The http or https proxy to use when
                           making requests. Can also be set
                           using LOKI_HTTP_PROXY_URL env var.
                           ($LOKI_HTTP_PROXY_URL)
      --[no-]compress      Request that Loki compress returned
                           data in transit. Can also be set
                           using LOKI_HTTP_COMPRESSION env var.
                           ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy      Use ProxyFromEnvironment to use net/http
                           ProxyFromEnvironment configuration, eg HTTP_PROXY
                           ($LOKI_ENV_PROXY)
      --since=1h           Lookback window.
      --from=FROM          Start looking for logs at this absolute time
                           (inclusive)
      --to=TO              Stop looking for logs at this absolute time
                           (exclusive)
      --[no-]analyze-labels   Printout a summary of labels including count of
                           label value combinations, useful for debugging
                           high cardinality series

Args:
  <matcher>  eg '{foo="bar",baz=~".*blip"}'
```

`stats` command reference

The output of `logcli help stats`:

```
usage: logcli stats [<flags>] <query>

Run a stats query.

The "stats" command will take the provided query and return statistics from the
```

```
index on how much data is contained in the matching stream(s). This only works
against Loki instances using the TSDB index format.

By default we look over the last hour of data; use --since to modify or provide
specific start and end times with --from and --to respectively.

Notice that when using --from and --to then ensure to use RFC3339Nano time
format, but without timezone at the end. The local timezone will be added
automatically or if using --timezone flag.

Example:

  logcli stats
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     'my-query'



Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
  -o, --output=default        Specify output mode [default, raw, jsonl].
                              raw suppresses log labels and timestamp.
  -z, --timezone=Local        Specify the timezone to use when formatting output
                              timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                              Specify the format of timestamps in the default
                              output mode [rfc3339, rfc3339nano, rfc822z,
                              rfc1123z, stampmicro, stampmilli, stampnano,
                              unixdate]
      --cpuprofile=""         Specify the location for writing a CPU profile.
      --memprofile=""         Specify the location for writing a memory profile.
      --[no-]stdin            Take input logs from stdin
      --addr="http://localhost:3100"
                              Server address. Can also be set using LOKI_ADDR
                              env var. ($LOKI_ADDR)
```

```
--username=""           Username for HTTP basic auth. Can also be set
                        using LOKI_USERNAME env var. ($LOKI_USERNAME)
--password=""           Password for HTTP basic auth. Can also be set
                        using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
--ca-cert=""            Path to the server Certificate Authority.
                        Can also be set using LOKI_CA_CERT_PATH env var.
                        ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                        be set using LOKI_TLS_SKIP_VERIFY env var.
                        ($LOKI_TLS_SKIP_VERIFY)
--cert=""               Path to the client certificate. Can also
                        be set using LOKI_CLIENT_CERT_PATH env var.
                        ($LOKI_CLIENT_CERT_PATH)
--key=""                Path to the client certificate key. Can also
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
--org-id=""             adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
--query-tags=""         adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
--[no-]nocache          adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
--bearer-token=""       adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
--bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
                        ($LOKI_BEARER_TOKEN_FILE)
--retries=0             How many times to retry each query when
                        getting an error response from Loki. Can also
                        be set using LOKI_CLIENT_RETRIES env var.
                        ($LOKI_CLIENT_RETRIES)
```

```
      --min-backoff=0         Minimum backoff time between retries. Can also
                              be set using LOKI_CLIENT_MIN_BACKOFF env var.
                              ($LOKI_CLIENT_MIN_BACKOFF)
      --max-backoff=0         Maximum backoff time between retries. Can also
                              be set using LOKI_CLIENT_MAX_BACKOFF env var.
                              ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                              The authorization header used. Can also
                              be set using LOKI_AUTH_HEADER env var.
                              ($LOKI_AUTH_HEADER)
      --proxy-url=""          The http or https proxy to use when
                              making requests. Can also be set
                              using LOKI_HTTP_PROXY_URL env var.
                              ($LOKI_HTTP_PROXY_URL)
      --[no-]compress         Request that Loki compress returned
                              data in transit. Can also be set
                              using LOKI_HTTP_COMPRESSION env var.
                              ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy         Use ProxyFromEnvironment to use net/http
                              ProxyFromEnvironment configuration, eg HTTP_PROXY
                              ($LOKI_ENV_PROXY)
      --since=1h              Lookback window.
      --from=FROM             Start looking for logs at this absolute time
                              (inclusive)
      --to=TO                 Stop looking for logs at this absolute time
                              (exclusive)

Args:
  <query>  eg '{foo="bar",baz=~".*blip"} |~ ".*error.*"'
```

`volume` command reference

The output of `logcli help volume`:

```
usage: logcli volume [<flags>] <query>

Run a volume query.

The "volume" command will take the provided label selector(s) and return
```

```
aggregate volumes for series matching those volumes. This only works against
Loki instances using the TSDB index format.

By default we look over the last hour of data; use --since to modify or provide
specific start and end times with --from and --to respectively.

Notice that when using --from and --to then ensure to use RFC3339Nano time
format, but without timezone at the end. The local timezone will be added
automatically or if using --timezone flag.

Example:

  logcli volume
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     'my-query'



Flags:
      --[no-]help               Show context-sensitive help (also try
                                --help-long and --help-man).
      --[no-]version            Show application version.
  -q, --[no-]quiet              Suppress query metadata
      --[no-]stats              Show query statistics
  -o, --output=default          Specify output mode [default, raw, jsonl].
                                raw suppresses log labels and timestamp.
  -z, --timezone=Local          Specify the timezone to use when formatting
                                output timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                                Specify the format of timestamps in the default
                                output mode [rfc3339, rfc3339nano, rfc822z,
                                rfc1123z, stampmicro, stampmilli, stampnano,
                                unixdate]
      --cpuprofile=""           Specify the location for writing a CPU profile.
      --memprofile=""           Specify the location for writing a memory
                                profile.
      --[no-]stdin              Take input logs from stdin
      --addr="http://localhost:3100"
                                Server address. Can also be set using LOKI_ADDR
```

```
                          env var. ($LOKI_ADDR)
    --username=""         Username for HTTP basic auth. Can also be set
                          using LOKI_USERNAME env var. ($LOKI_USERNAME)
    --password=""         Password for HTTP basic auth. Can also be set
                          using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
    --ca-cert=""          Path to the server Certificate Authority.
                          Can also be set using LOKI_CA_CERT_PATH env var.
                          ($LOKI_CA_CERT_PATH)
    --[no-]tls-skip-verify    Server certificate TLS skip verify. Can also
                          be set using LOKI_TLS_SKIP_VERIFY env var.
                          ($LOKI_TLS_SKIP_VERIFY)
    --cert=""             Path to the client certificate. Can also
                          be set using LOKI_CLIENT_CERT_PATH env var.
                          ($LOKI_CLIENT_CERT_PATH)
    --key=""              Path to the client certificate key. Can also
                          be set using LOKI_CLIENT_KEY_PATH env var.
                          ($LOKI_CLIENT_KEY_PATH)
    --org-id=""           adds X-Scope-OrgID to API requests for
                          representing tenant ID. Useful for requesting
                          tenant data when bypassing an auth gateway.
                          Can also be set using LOKI_ORG_ID env var.
                          ($LOKI_ORG_ID)
    --query-tags=""       adds X-Query-Tags http header to API requests.
                          This header value will be part of `metrics.go`
                          statistics. Useful for tracking the query.
                          Can also be set using LOKI_QUERY_TAGS env var.
                          ($LOKI_QUERY_TAGS)
    --[no-]nocache        adds Cache-Control: no-cache http header to API
                          requests. Can also be set using LOKI_NO_CACHE
                          env var. ($LOKI_NO_CACHE)
    --bearer-token=""     adds the Authorization header to API
                          requests for authentication purposes.
                          Can also be set using LOKI_BEARER_TOKEN env var.
                          ($LOKI_BEARER_TOKEN)
    --bearer-token-file=""    adds the Authorization header to API requests
                          for authentication purposes. Can also be
                          set using LOKI_BEARER_TOKEN_FILE env var.
                          ($LOKI_BEARER_TOKEN_FILE)
    --retries=0           How many times to retry each query when
                          getting an error response from Loki. Can also
```

```
                              be set using LOKI_CLIENT_RETRIES env var.
                              ($LOKI_CLIENT_RETRIES)
      --min-backoff=0         Minimum backoff time between retries. Can also
                              be set using LOKI_CLIENT_MIN_BACKOFF env var.
                              ($LOKI_CLIENT_MIN_BACKOFF)
      --max-backoff=0         Maximum backoff time between retries. Can also
                              be set using LOKI_CLIENT_MAX_BACKOFF env var.
                              ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                              The authorization header used. Can also
                              be set using LOKI_AUTH_HEADER env var.
                              ($LOKI_AUTH_HEADER)
      --proxy-url=""          The http or https proxy to use when
                              making requests. Can also be set
                              using LOKI_HTTP_PROXY_URL env var.
                              ($LOKI_HTTP_PROXY_URL)
      --[no-]compress         Request that Loki compress returned
                              data in transit. Can also be set
                              using LOKI_HTTP_COMPRESSION env var.
                              ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy         Use ProxyFromEnvironment to use net/http
                              ProxyFromEnvironment configuration, eg
                              HTTP_PROXY ($LOKI_ENV_PROXY)
      --since=1h              Lookback window.
      --from=FROM             Start looking for logs at this absolute time
                              (inclusive)
      --to=TO                 Stop looking for logs at this absolute time
                              (exclusive)
      --limit=30              Limit on number of series to return volumes for.
      --targetLabels=TARGETLABELS ...
                              List of labels to aggregate results into.
      --[no-]aggregateByLabels  Whether to aggregate results by label name only.

Args:
  <query>  eg '{foo="bar",baz=~".*blip"}
```

`volume_range` command reference

The output of `logcli help volume_range`:

```
usage: logcli volume_range [<flags>] <query>

Run a volume query and return timeseries data.

The "volume_range" command will take the provided label selector(s) and return
aggregate volumes for series matching those volumes, aggregated into buckets
according to the step value. This only works against Loki instances using the
TSDB index format.

By default we look over the last hour of data; use --since to modify or provide
specific start and end times with --from and --to respectively.

Notice that when using --from and --to then ensure to use RFC3339Nano time
format, but without timezone at the end. The local timezone will be added
automatically or if using --timezone flag.

Example:

  logcli volume_range
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
      --step=1h
     'my-query'



Flags:
      --[no-]help              Show context-sensitive help (also try
                               --help-long and --help-man).
      --[no-]version           Show application version.
  -q, --[no-]quiet             Suppress query metadata
      --[no-]stats             Show query statistics
  -o, --output=default         Specify output mode [default, raw, jsonl].
                               raw suppresses log labels and timestamp.
  -z, --timezone=Local         Specify the timezone to use when formatting
                               output timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                               Specify the format of timestamps in the default
                               output mode [rfc3339, rfc3339nano, rfc822z,
```

```
                        rfc1123z, stampmicro, stampmilli, stampnano,
                        unixdate]
--cpuprofile=""         Specify the location for writing a CPU profile.
--memprofile=""         Specify the location for writing a memory
                        profile.
--[no-]stdin            Take input logs from stdin
--addr="http://localhost:3100"
                        Server address. Can also be set using LOKI_ADDR
                        env var. ($LOKI_ADDR)
--username=""           Username for HTTP basic auth. Can also be set
                        using LOKI_USERNAME env var. ($LOKI_USERNAME)
--password=""           Password for HTTP basic auth. Can also be set
                        using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
--ca-cert=""            Path to the server Certificate Authority.
                        Can also be set using LOKI_CA_CERT_PATH env var.
                        ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                        be set using LOKI_TLS_SKIP_VERIFY env var.
                        ($LOKI_TLS_SKIP_VERIFY)
--cert=""               Path to the client certificate. Can also
                        be set using LOKI_CLIENT_CERT_PATH env var.
                        ($LOKI_CLIENT_CERT_PATH)
--key=""                Path to the client certificate key. Can also
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
--org-id=""             adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
--query-tags=""         adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
--[no-]nocache          adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE
                        env var. ($LOKI_NO_CACHE)
--bearer-token=""       adds the Authorization header to API
                        requests for authentication purposes.
```

```
                          Can also be set using LOKI_BEARER_TOKEN env var.
                          ($LOKI_BEARER_TOKEN)
    --bearer-token-file=""    adds the Authorization header to API requests
                          for authentication purposes. Can also be
                          set using LOKI_BEARER_TOKEN_FILE env var.
                          ($LOKI_BEARER_TOKEN_FILE)
    --retries=0               How many times to retry each query when
                          getting an error response from Loki. Can also
                          be set using LOKI_CLIENT_RETRIES env var.
                          ($LOKI_CLIENT_RETRIES)
    --min-backoff=0           Minimum backoff time between retries. Can also
                          be set using LOKI_CLIENT_MIN_BACKOFF env var.
                          ($LOKI_CLIENT_MIN_BACKOFF)
    --max-backoff=0           Maximum backoff time between retries. Can also
                          be set using LOKI_CLIENT_MAX_BACKOFF env var.
                          ($LOKI_CLIENT_MAX_BACKOFF)
    --auth-header="Authorization"
                          The authorization header used. Can also
                          be set using LOKI_AUTH_HEADER env var.
                          ($LOKI_AUTH_HEADER)
    --proxy-url=""            The http or https proxy to use when
                          making requests. Can also be set
                          using LOKI_HTTP_PROXY_URL env var.
                          ($LOKI_HTTP_PROXY_URL)
    --[no-]compress           Request that Loki compress returned
                          data in transit. Can also be set
                          using LOKI_HTTP_COMPRESSION env var.
                          ($LOKI_HTTP_COMPRESSION)
    --[no-]envproxy           Use ProxyFromEnvironment to use net/http
                          ProxyFromEnvironment configuration, eg
                          HTTP_PROXY ($LOKI_ENV_PROXY)
    --since=1h                Lookback window.
    --from=FROM               Start looking for logs at this absolute time
                          (inclusive)
    --to=TO                   Stop looking for logs at this absolute time
                          (exclusive)
    --limit=30                Limit on number of series to return volumes for.
    --targetLabels=TARGETLABELS ...
                          List of labels to aggregate results into.
    --[no-]aggregateByLabels  Whether to aggregate results by label name only.
```

```
      --step=1h                Query resolution step width, roll up volumes
                               into buckets cover step time each.

Args:
  <query>  eg '{foo="bar",baz=~".*blip"}
```

`detected-fields` command reference

The output of `logcli help detected-fields`:

```
usage: logcli detected-fields [<flags>] <query> [<field>]

Run a query for detected fields..

The "detected-fields" command will return information about fields detected
using either the "logfmt" or "json" parser against the log lines returned by the
provided query for the provided time range.

The "detected-fields" command will output extra information about the query
and its results, such as the API URL, set of common labels, and set of excluded
labels. This extra information can be suppressed with the --quiet flag.

By default we look over the last hour of data; use --since to modify or provide
specific start and end times with --from and --to respectively.

Notice that when using --from and --to then ensure to use RFC3339Nano time
format, but without timezone at the end. The local timezone will be added
automatically or if using --timezone flag.

Example:

  logcli detected-fields
     --timezone=UTC
     --from="2021-01-19T10:00:00Z"
     --to="2021-01-19T20:00:00Z"
     --output=jsonl
     'my-query'

The output is limited to 100 fields by default; use --field-limit to increase.
```

```
The query is limited to processing 1000 lines per subquery; use --line-limit to
increase.


Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
  -o, --output=default        Specify output mode [default, raw, jsonl].
                              raw suppresses log labels and timestamp.
  -z, --timezone=Local        Specify the timezone to use when formatting output
                              timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                              Specify the format of timestamps in the default
                              output mode [rfc3339, rfc3339nano, rfc822z,
                              rfc1123z, stampmicro, stampmilli, stampnano,
                              unixdate]
      --cpuprofile=""         Specify the location for writing a CPU profile.
      --memprofile=""         Specify the location for writing a memory profile.
      --[no-]stdin            Take input logs from stdin
      --addr="http://localhost:3100"
                              Server address. Can also be set using LOKI_ADDR
                              env var. ($LOKI_ADDR)
      --username=""           Username for HTTP basic auth. Can also be set
                              using LOKI_USERNAME env var. ($LOKI_USERNAME)
      --password=""           Password for HTTP basic auth. Can also be set
                              using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
      --ca-cert=""            Path to the server Certificate Authority.
                              Can also be set using LOKI_CA_CERT_PATH env var.
                              ($LOKI_CA_CERT_PATH)
      --[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                              be set using LOKI_TLS_SKIP_VERIFY env var.
                              ($LOKI_TLS_SKIP_VERIFY)
      --cert=""               Path to the client certificate. Can also
                              be set using LOKI_CLIENT_CERT_PATH env var.
                              ($LOKI_CLIENT_CERT_PATH)
      --key=""                Path to the client certificate key. Can also
                              be set using LOKI_CLIENT_KEY_PATH env var.
```

```
                        ($LOKI_CLIENT_KEY_PATH)
    --org-id=""         adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
    --query-tags=""     adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
    --[no-]nocache      adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
    --bearer-token=""   adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
    --bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
                        ($LOKI_BEARER_TOKEN_FILE)
    --retries=0         How many times to retry each query when
                        getting an error response from Loki. Can also
                        be set using LOKI_CLIENT_RETRIES env var.
                        ($LOKI_CLIENT_RETRIES)
    --min-backoff=0     Minimum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MIN_BACKOFF env var.
                        ($LOKI_CLIENT_MIN_BACKOFF)
    --max-backoff=0     Maximum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MAX_BACKOFF env var.
                        ($LOKI_CLIENT_MAX_BACKOFF)
    --auth-header="Authorization"
                        The authorization header used. Can also
                        be set using LOKI_AUTH_HEADER env var.
                        ($LOKI_AUTH_HEADER)
    --proxy-url=""      The http or https proxy to use when
                        making requests. Can also be set
                        using LOKI_HTTP_PROXY_URL env var.
                        ($LOKI_HTTP_PROXY_URL)
    --[no-]compress     Request that Loki compress returned
```

```
                           data in transit. Can also be set
                           using LOKI_HTTP_COMPRESSION env var.
                           ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy      Use ProxyFromEnvironment to use net/http
                           ProxyFromEnvironment configuration, eg HTTP_PROXY
                           ($LOKI_ENV_PROXY)
      --limit=100          Limit on number of fields or values to return.
      --line-limit=1000    Limit the number of lines each subquery is allowed
                           to process.
      --since=1h           Lookback window.
      --from=FROM          Start looking for logs at this absolute time
                           (inclusive)
      --to=TO              Stop looking for logs at this absolute time
                           (exclusive)
      --step=10s           Query resolution step width, for metric queries.
                           Evaluate the query at the specified step over the
                           time range.

Args:
  <query>    eg '{foo="bar",baz=~".*blip"} |~ ".*error.*"'
  [<field>]  The name of the field.
```

`delete` command reference

The output of `logcli help delete`:

```
usage: logcli delete <command> [<args> ...]

Manage log deletion requests.


Flags:
      --[no-]help          Show context-sensitive help (also try --help-long
                           and --help-man).
      --[no-]version       Show application version.
  -q, --[no-]quiet         Suppress query metadata
      --[no-]stats         Show query statistics
  -o, --output=default     Specify output mode [default, raw, jsonl].
                           raw suppresses log labels and timestamp.
```

```
-z, --timezone=Local        Specify the timezone to use when formatting output
                            timestamps [Local, UTC]
    --output-timestamp-format=rfc3339
                            Specify the format of timestamps in the default
                            output mode [rfc3339, rfc3339nano, rfc822z,
                            rfc1123z, stampmicro, stampmilli, stampnano,
                            unixdate]
    --cpuprofile=""         Specify the location for writing a CPU profile.
    --memprofile=""         Specify the location for writing a memory profile.
    --[no-]stdin            Take input logs from stdin
    --addr="http://localhost:3100"
                            Server address. Can also be set using LOKI_ADDR
                            env var. ($LOKI_ADDR)
    --username=""           Username for HTTP basic auth. Can also be set
                            using LOKI_USERNAME env var. ($LOKI_USERNAME)
    --password=""           Password for HTTP basic auth. Can also be set
                            using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
    --ca-cert=""            Path to the server Certificate Authority.
                            Can also be set using LOKI_CA_CERT_PATH env var.
                            ($LOKI_CA_CERT_PATH)
    --[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                            be set using LOKI_TLS_SKIP_VERIFY env var.
                            ($LOKI_TLS_SKIP_VERIFY)
    --cert=""               Path to the client certificate. Can also
                            be set using LOKI_CLIENT_CERT_PATH env var.
                            ($LOKI_CLIENT_CERT_PATH)
    --key=""                Path to the client certificate key. Can also
                            be set using LOKI_CLIENT_KEY_PATH env var.
                            ($LOKI_CLIENT_KEY_PATH)
    --org-id=""             adds X-Scope-OrgID to API requests for
                            representing tenant ID. Useful for requesting
                            tenant data when bypassing an auth gateway.
                            Can also be set using LOKI_ORG_ID env var.
                            ($LOKI_ORG_ID)
    --query-tags=""         adds X-Query-Tags http header to API requests.
                            This header value will be part of `metrics.go`
                            statistics. Useful for tracking the query.
                            Can also be set using LOKI_QUERY_TAGS env var.
                            ($LOKI_QUERY_TAGS)
    --[no-]nocache          adds Cache-Control: no-cache http header to API
```

```
                           requests. Can also be set using LOKI_NO_CACHE env
                           var. ($LOKI_NO_CACHE)
      --bearer-token=""    adds the Authorization header to API requests for
                           authentication purposes. Can also be set using
                           LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
      --bearer-token-file=""  adds the Authorization header to API requests
                           for authentication purposes. Can also be
                           set using LOKI_BEARER_TOKEN_FILE env var.
                           ($LOKI_BEARER_TOKEN_FILE)
      --retries=0          How many times to retry each query when
                           getting an error response from Loki. Can also
                           be set using LOKI_CLIENT_RETRIES env var.
                           ($LOKI_CLIENT_RETRIES)
      --min-backoff=0      Minimum backoff time between retries. Can also
                           be set using LOKI_CLIENT_MIN_BACKOFF env var.
                           ($LOKI_CLIENT_MIN_BACKOFF)
      --max-backoff=0      Maximum backoff time between retries. Can also
                           be set using LOKI_CLIENT_MAX_BACKOFF env var.
                           ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                           The authorization header used. Can also
                           be set using LOKI_AUTH_HEADER env var.
                           ($LOKI_AUTH_HEADER)
      --proxy-url=""       The http or https proxy to use when
                           making requests. Can also be set
                           using LOKI_HTTP_PROXY_URL env var.
                           ($LOKI_HTTP_PROXY_URL)
      --[no-]compress      Request that Loki compress returned
                           data in transit. Can also be set
                           using LOKI_HTTP_COMPRESSION env var.
                           ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy      Use ProxyFromEnvironment to use net/http
                           ProxyFromEnvironment configuration, eg HTTP_PROXY
                           ($LOKI_ENV_PROXY)

Subcommands:
delete create [<flags>] <query>
    Create a new log deletion request.

    The "delete create" command creates a new log deletion request for the
```

```
    specified query and time range.

    Example:

      logcli delete create '{job="app"}' --from="2023-01-01T00:00:00Z" --to="2023-01-
02T00:00:00Z"

delete list
    List existing log deletion requests.

    The "delete list" command lists all existing log deletion requests for the
    tenant.

    Example:

      logcli delete list
      logcli delete list --output=json

delete cancel --request-id=REQUEST-ID [<flags>]
    Cancel a log deletion request.

    The "delete cancel" command cancels an existing log deletion request by ID.

    Example:

      logcli delete cancel --request-id="abc123"
      logcli delete cancel --request-id="abc123" --force
```

`delete create` command reference

The output of `logcli help delete create`:

```
usage: logcli delete create [<flags>] <query>

Create a new log deletion request.

The "delete create" command creates a new log deletion request for the specified
query and time range.
```

```
Example:

  logcli delete create '{job="app"}' --from="2023-01-01T00:00:00Z" --to="2023-01-02T00:00:00Z"


Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
  -o, --output=default        Specify output mode [default, raw, jsonl].
                              raw suppresses log labels and timestamp.
  -z, --timezone=Local        Specify the timezone to use when formatting output
                              timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                              Specify the format of timestamps in the default
                              output mode [rfc3339, rfc3339nano, rfc822z,
                              rfc1123z, stampmicro, stampmilli, stampnano,
                              unixdate]
      --cpuprofile=""         Specify the location for writing a CPU profile.
      --memprofile=""         Specify the location for writing a memory profile.
      --[no-]stdin            Take input logs from stdin
      --addr="http://localhost:3100"
                              Server address. Can also be set using LOKI_ADDR
                              env var. ($LOKI_ADDR)
      --username=""           Username for HTTP basic auth. Can also be set
                              using LOKI_USERNAME env var. ($LOKI_USERNAME)
      --password=""           Password for HTTP basic auth. Can also be set
                              using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
      --ca-cert=""            Path to the server Certificate Authority.
                              Can also be set using LOKI_CA_CERT_PATH env var.
                              ($LOKI_CA_CERT_PATH)
      --[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                              be set using LOKI_TLS_SKIP_VERIFY env var.
                              ($LOKI_TLS_SKIP_VERIFY)
      --cert=""               Path to the client certificate. Can also
                              be set using LOKI_CLIENT_CERT_PATH env var.
                              ($LOKI_CLIENT_CERT_PATH)
      --key=""                Path to the client certificate key. Can also
```

```
                        be set using LOKI_CLIENT_KEY_PATH env var.
                        ($LOKI_CLIENT_KEY_PATH)
    --org-id=""         adds X-Scope-OrgID to API requests for
                        representing tenant ID. Useful for requesting
                        tenant data when bypassing an auth gateway.
                        Can also be set using LOKI_ORG_ID env var.
                        ($LOKI_ORG_ID)
    --query-tags=""     adds X-Query-Tags http header to API requests.
                        This header value will be part of `metrics.go`
                        statistics. Useful for tracking the query.
                        Can also be set using LOKI_QUERY_TAGS env var.
                        ($LOKI_QUERY_TAGS)
    --[no-]nocache      adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
    --bearer-token=""   adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
    --bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
                        ($LOKI_BEARER_TOKEN_FILE)
    --retries=0         How many times to retry each query when
                        getting an error response from Loki. Can also
                        be set using LOKI_CLIENT_RETRIES env var.
                        ($LOKI_CLIENT_RETRIES)
    --min-backoff=0     Minimum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MIN_BACKOFF env var.
                        ($LOKI_CLIENT_MIN_BACKOFF)
    --max-backoff=0     Maximum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MAX_BACKOFF env var.
                        ($LOKI_CLIENT_MAX_BACKOFF)
    --auth-header="Authorization"
                        The authorization header used. Can also
                        be set using LOKI_AUTH_HEADER env var.
                        ($LOKI_AUTH_HEADER)
    --proxy-url=""      The http or https proxy to use when
                        making requests. Can also be set
                        using LOKI_HTTP_PROXY_URL env var.
                        ($LOKI_HTTP_PROXY_URL)
```

```
      --[no-]compress         Request that Loki compress returned
                              data in transit. Can also be set
                              using LOKI_HTTP_COMPRESSION env var.
                              ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy         Use ProxyFromEnvironment to use net/http
                              ProxyFromEnvironment configuration, eg HTTP_PROXY
                              ($LOKI_ENV_PROXY)
      --since=1h              Lookback window.
      --from=FROM             Start time for deletion (inclusive)
      --to=TO                 End time for deletion (exclusive)
      --max-interval=MAX-INTERVAL
                              Maximum time interval for delete request

Args:
  <query>  LogQL query to match log lines for deletion (e.g. '{job="app"}')
```

`delete list` command reference

The output of `logcli help delete list`:

```
usage: logcli delete list

List existing log deletion requests.

The "delete list" command lists all existing log deletion requests for the
tenant.

Example:

  logcli delete list
  logcli delete list --output=json



Flags:
      --[no-]help             Show context-sensitive help (also try --help-long
                              and --help-man).
      --[no-]version          Show application version.
  -q, --[no-]quiet            Suppress query metadata
      --[no-]stats            Show query statistics
```

```
-o, --output=default        Specify output mode [default, raw, jsonl].
                            raw suppresses log labels and timestamp.
-z, --timezone=Local        Specify the timezone to use when formatting output
                            timestamps [Local, UTC]
    --output-timestamp-format=rfc3339
                            Specify the format of timestamps in the default
                            output mode [rfc3339, rfc3339nano, rfc822z,
                            rfc1123z, stampmicro, stampmilli, stampnano,
                            unixdate]
    --cpuprofile=""         Specify the location for writing a CPU profile.
    --memprofile=""         Specify the location for writing a memory profile.
    --[no-]stdin            Take input logs from stdin
    --addr="http://localhost:3100"
                            Server address. Can also be set using LOKI_ADDR
                            env var. ($LOKI_ADDR)
    --username=""           Username for HTTP basic auth. Can also be set
                            using LOKI_USERNAME env var. ($LOKI_USERNAME)
    --password=""           Password for HTTP basic auth. Can also be set
                            using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
    --ca-cert=""            Path to the server Certificate Authority.
                            Can also be set using LOKI_CA_CERT_PATH env var.
                            ($LOKI_CA_CERT_PATH)
    --[no-]tls-skip-verify  Server certificate TLS skip verify. Can also
                            be set using LOKI_TLS_SKIP_VERIFY env var.
                            ($LOKI_TLS_SKIP_VERIFY)
    --cert=""               Path to the client certificate. Can also
                            be set using LOKI_CLIENT_CERT_PATH env var.
                            ($LOKI_CLIENT_CERT_PATH)
    --key=""                Path to the client certificate key. Can also
                            be set using LOKI_CLIENT_KEY_PATH env var.
                            ($LOKI_CLIENT_KEY_PATH)
    --org-id=""             adds X-Scope-OrgID to API requests for
                            representing tenant ID. Useful for requesting
                            tenant data when bypassing an auth gateway.
                            Can also be set using LOKI_ORG_ID env var.
                            ($LOKI_ORG_ID)
    --query-tags=""         adds X-Query-Tags http header to API requests.
                            This header value will be part of `metrics.go`
                            statistics. Useful for tracking the query.
                            Can also be set using LOKI_QUERY_TAGS env var.
```

```
                        ($LOKI_QUERY_TAGS)
    --[no-]nocache      adds Cache-Control: no-cache http header to API
                        requests. Can also be set using LOKI_NO_CACHE env
                        var. ($LOKI_NO_CACHE)
    --bearer-token=""   adds the Authorization header to API requests for
                        authentication purposes. Can also be set using
                        LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
    --bearer-token-file=""  adds the Authorization header to API requests
                        for authentication purposes. Can also be
                        set using LOKI_BEARER_TOKEN_FILE env var.
                        ($LOKI_BEARER_TOKEN_FILE)
    --retries=0         How many times to retry each query when
                        getting an error response from Loki. Can also
                        be set using LOKI_CLIENT_RETRIES env var.
                        ($LOKI_CLIENT_RETRIES)
    --min-backoff=0     Minimum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MIN_BACKOFF env var.
                        ($LOKI_CLIENT_MIN_BACKOFF)
    --max-backoff=0     Maximum backoff time between retries. Can also
                        be set using LOKI_CLIENT_MAX_BACKOFF env var.
                        ($LOKI_CLIENT_MAX_BACKOFF)
    --auth-header="Authorization"
                        The authorization header used. Can also
                        be set using LOKI_AUTH_HEADER env var.
                        ($LOKI_AUTH_HEADER)
    --proxy-url=""      The http or https proxy to use when
                        making requests. Can also be set
                        using LOKI_HTTP_PROXY_URL env var.
                        ($LOKI_HTTP_PROXY_URL)
    --[no-]compress     Request that Loki compress returned
                        data in transit. Can also be set
                        using LOKI_HTTP_COMPRESSION env var.
                        ($LOKI_HTTP_COMPRESSION)
    --[no-]envproxy     Use ProxyFromEnvironment to use net/http
                        ProxyFromEnvironment configuration, eg HTTP_PROXY
                        ($LOKI_ENV_PROXY)
```

`delete cancel` command reference

The output of `logcli help delete cancel`:

```
usage: logcli delete cancel --request-id=REQUEST-ID [<flags>]

Cancel a log deletion request.

The "delete cancel" command cancels an existing log deletion request by ID.

Example:

  logcli delete cancel --request-id="abc123"
  logcli delete cancel --request-id="abc123" --force



Flags:
      --[no-]help              Show context-sensitive help (also try --help-long
                               and --help-man).
      --[no-]version           Show application version.
  -q, --[no-]quiet             Suppress query metadata
      --[no-]stats             Show query statistics
  -o, --output=default         Specify output mode [default, raw, jsonl].
                               raw suppresses log labels and timestamp.
  -z, --timezone=Local         Specify the timezone to use when formatting
                               output timestamps [Local, UTC]
      --output-timestamp-format=rfc3339
                               Specify the format of timestamps in the default
                               output mode [rfc3339, rfc3339nano, rfc822z,
                               rfc1123z, stampmicro, stampmilli, stampnano,
                               unixdate]
      --cpuprofile=""          Specify the location for writing a CPU profile.
      --memprofile=""          Specify the location for writing a memory
                               profile.
      --[no-]stdin             Take input logs from stdin
      --addr="http://localhost:3100"
                               Server address. Can also be set using LOKI_ADDR
                               env var. ($LOKI_ADDR)
      --username=""            Username for HTTP basic auth. Can also be set
                               using LOKI_USERNAME env var. ($LOKI_USERNAME)
      --password=""            Password for HTTP basic auth. Can also be set
                               using LOKI_PASSWORD env var. ($LOKI_PASSWORD)
```

```
--ca-cert=""              Path to the server Certificate Authority.
                          Can also be set using LOKI_CA_CERT_PATH env var.
                          ($LOKI_CA_CERT_PATH)
--[no-]tls-skip-verify    Server certificate TLS skip verify. Can also
                          be set using LOKI_TLS_SKIP_VERIFY env var.
                          ($LOKI_TLS_SKIP_VERIFY)
--cert=""                 Path to the client certificate. Can also
                          be set using LOKI_CLIENT_CERT_PATH env var.
                          ($LOKI_CLIENT_CERT_PATH)
--key=""                  Path to the client certificate key. Can also
                          be set using LOKI_CLIENT_KEY_PATH env var.
                          ($LOKI_CLIENT_KEY_PATH)
--org-id=""               adds X-Scope-OrgID to API requests for
                          representing tenant ID. Useful for requesting
                          tenant data when bypassing an auth gateway.
                          Can also be set using LOKI_ORG_ID env var.
                          ($LOKI_ORG_ID)
--query-tags=""           adds X-Query-Tags http header to API requests.
                          This header value will be part of `metrics.go`
                          statistics. Useful for tracking the query.
                          Can also be set using LOKI_QUERY_TAGS env var.
                          ($LOKI_QUERY_TAGS)
--[no-]nocache            adds Cache-Control: no-cache http header to API
                          requests. Can also be set using LOKI_NO_CACHE env
                          var. ($LOKI_NO_CACHE)
--bearer-token=""         adds the Authorization header to API requests for
                          authentication purposes. Can also be set using
                          LOKI_BEARER_TOKEN env var. ($LOKI_BEARER_TOKEN)
--bearer-token-file=""    adds the Authorization header to API requests
                          for authentication purposes. Can also be
                          set using LOKI_BEARER_TOKEN_FILE env var.
                          ($LOKI_BEARER_TOKEN_FILE)
--retries=0               How many times to retry each query when
                          getting an error response from Loki. Can also
                          be set using LOKI_CLIENT_RETRIES env var.
                          ($LOKI_CLIENT_RETRIES)
--min-backoff=0           Minimum backoff time between retries. Can also
                          be set using LOKI_CLIENT_MIN_BACKOFF env var.
                          ($LOKI_CLIENT_MIN_BACKOFF)
--max-backoff=0           Maximum backoff time between retries. Can also
```

```
                          be set using LOKI_CLIENT_MAX_BACKOFF env var.
                          ($LOKI_CLIENT_MAX_BACKOFF)
      --auth-header="Authorization"
                          The authorization header used. Can also
                          be set using LOKI_AUTH_HEADER env var.
                          ($LOKI_AUTH_HEADER)
      --proxy-url=""      The http or https proxy to use when
                          making requests. Can also be set
                          using LOKI_HTTP_PROXY_URL env var.
                          ($LOKI_HTTP_PROXY_URL)
      --[no-]compress     Request that Loki compress returned
                          data in transit. Can also be set
                          using LOKI_HTTP_COMPRESSION env var.
                          ($LOKI_HTTP_COMPRESSION)
      --[no-]envproxy     Use ProxyFromEnvironment to use net/http
                          ProxyFromEnvironment configuration, eg HTTP_PROXY
                          ($LOKI_ENV_PROXY)
      --request-id=REQUEST-ID  ID of the delete request to cancel
      --[no-]force        Force cancellation of partially completed request
```

Use `--stdin` to query locally

You can use the logcli `–stdin` argument to run a command against a log file on your local machine, instead of a Loki instance. This lets you use LogQL to query a local log file without having to load the file into Loki, for example if you have downloaded a log file and want to query it outside of Loki.

If you have log files in your local machine, and just want to run some LogQL queries against those log files, `--stdin` flag can help.

You may use `stdin` flag to do the following:

- Use as a quick way to test or validate a LogQL expression against some log data.
- Learn the basics of LogQL with just local log files and the `logcli` tool (without needing to set up Loki servers, Grafana, etc.).
- Enable troubleshooting by letting you run queries without accessing a Loki instance.
- Use LogQL to parse and extract data from a local log file without ingesting the data into Loki.
- Enable discussion on public forums, for example submitting questions and answers, and sharing LogQL expressions.

Notes on `stdin` usage

1. The `--limit` flag doesn't have any meaning when using `--stdin` (use pager like `less` for that).
2. Be aware there are no **labels** when using `--stdin`. So the stream selector in the query is optional, for example, just `|="timeout"|logfmt|level="error"` is same as `{foo="bar"}|="timeout|logfmt|level="error"`.

{{< admonition type="note" >}} Currently `stdin` doesn't support any type of metric queries. {{< /admonition >}}

`stdin` examples

- Line filter - `cat mylog.log | logcli --stdin query '|="too many open connections"'`
- Label matcher - `echo 'msg="timeout happened" level="warning"' | logcli --stdin query '|logfmt|level="warning"'`
- Different parsers (logfmt, json, pattern, regexp) - `cat mylog.log | logcli --stdin query '|pattern <ip> - - <_> "<method> <uri> <_>" <status> <size> <_> "<agent>" <_>'`
- Line formatters - `cat mylog.log | logcli --stdin query '|logfmt|line_format "{{.query}} {{.duration}}"'`

Batching

logcli sends queries to Loki in such a way that query results arrive in batches.

The `--limit` option for a `logcli query` command limits the total number of log lines that will be returned for a single query. When not set, `--limit` defaults to 30. The limit protects the user from overwhelming Loki in cases where the specified query would have returned a large number of log lines. The limit also protects the user from unexpectedly large responses.

Larger result sets can be batched for easier consumption. Use the `--batch` option to control the number of log line results that are returned in each batch. When not set, `--batch` defaults to 1000.

Setting a `--limit` value larger than the `--batch` value will cause the requests from logcli to Loki to be batched.

When you run a query in Loki, it will return up to a certain number of log lines. By default, this limit is 5000 lines. You can configure this server limit with the `limits_config.max_entries_limit_per_query` in Loki's configuration.

Batching lets you query for a results set that is larger than this server-side limit, as long as the `--batch` value is less than the server limit.

Query metadata is output to `stderr` for each batch. To suppress the output of the query metadata, set the `--quiet` option on the `logcli query` command line.

logcli example queries

Here are some examples of logcli.

Find all values for a label.

```
logcli labels job
```

```
https://logs-dev-ops-tools1.grafana.net/api/prom/label/job/values
loki-ops/consul
loki-ops/loki-gw
```

Print all labels and their unique values. This command is especially useful for finding high-cardinality labels in the index.

```
logcli series '{cluster="vinson"}' --analyze-labels
```

```
2024/10/31 13:46:25 https://logs-prod-
008.grafana.net/loki/api/v1/series?end=1730382385746344416&match=%7Bcluster%3D%22vinson%22%7D&
start=1730378785746344416
Total Streams:  10
Unique Labels:  10

Label Name       Unique Values  Found In Streams
service_name        8          10
pod                 7          7
job                 6          10
app_kubernetes_io_name  6          6
container           5          7
namespace           3          10
stream              2          7
flags               1          7
instance            1          3
cluster             1          10
```

Get all logs for a given stream

```
logcli query '{job="loki-ops/consul"}'
```

```
https://logs-dev-ops-tools1.grafana.net/api/prom/query?query=%7Bjob%3D%22loki-ops%2Fconsul%22%7D&limit=30&start=1529928228&end=1529931828&direction=backward&regexp=
Common labels: {job="loki-ops/consul", namespace="loki-ops"}
2018-06-25T12:52:09Z {instance="consul-8576459955-pl75w"} 2018/06/25 12:52:09 [INFO] raft: Snapshot to 475409 complete
2018-06-25T12:52:09Z {instance="consul-8576459955-pl75w"} 2018/06/25 12:52:09 [INFO] raft: Compacting logs from 456973 to 465169
```

Print all log streams for the given stream selector. This example shows all known label combinations that match your query.

```
logcli series -q --match='{namespace="loki",container_name="loki"}'
```

```
{app="loki", container_name="loki", controller_revision_hash="loki-57c9df47f4", filename="/var/log/pods/loki_loki-0_8ed03ded-bacb-4b13-a6fe-53a445a15887/loki/0.log", instance="loki-0", job="loki/loki", name="loki", namespace="loki", release="loki", statefulset_kubernetes_io_pod_name="loki-0", stream="stderr"}
```

Delete all logs for a specific job within a time range:

```
logcli delete create '{job="app"}' --from="2023-01-01T00:00:00Z" --to="2023-01-02T00:00:00Z"
```

List all deletion requests:

```
logcli delete list
```

Cancel a specific deletion request:

```
logcli delete cancel --request-id="abc123"
```

Troubleshoot logcli

Make sure that the version of Logcli you are using matches your Loki version. You can check your logcli version with the following command:

```
logcli –version
```

If you experience timeouts, you can update the following setting in your `logcli-config.yaml` file.

```
limits_config:
  query_timeout: 10m
```

□□□ #2
□ □□□ □□□ 2026-10-31 13:47:44 UTC
□ □□□ □□□ 2026-10-31 13:47:47 UTC