LOGZILLA DOCUMENTATION

Data Commands

LogZilla CLI data commands for archive creation, retention policies, rule management, trigger execution, dashboard import and export, and event lifecycle

Command Line Tools · Generated April 27, 2026 · logzilla.ai/docs/command-line-tools/data-commands

Data Management Commands

LogZilla provides command line tools for managing the complete data lifecycle, from ingestion and processing to archival and deletion. These commands handle data flow, storage optimization, event processing rules, and automated responses to events.

Data Storage and Lifecycle

Archive Management

Control data archiving to manage disk space and implement data retention policies. Archives move older data to compressed storage while maintaining searchability.

Create Archives

```
# Archive events older than N days
logzilla archives archive --expire-days 30

# Archive specific date range (rounded to full hours)
logzilla archives --round archive --ts-from "2024-01-01 00:00" --ts-to "2024-01-31 23:00"

# Round provided timestamps to full hours automatically
logzilla archives --round archive --expire-days 60
```

Migrate Old Archives

```
# Migrate archives to the latest format for direct querying
logzilla archives migrate --ts-from "2023-01-01 00:00" --ts-to "2023-12-31 23:00"
```

Remove Archives

```
# Remove archived data permanently (use with caution)
logzilla archives remove --ts-from "2022-01-01 00:00" --ts-to "2022-12-31 23:00"
```

Event Statistics and Monitoring

Monitor data volume, system performance, and storage utilization.

View Event Statistics

```
# View event statistics for a date range (print totals)
logzilla events stats --ts-from "2024-01-01 00:00" --ts-to "2024-01-31 23:59"

# Include full per-chunk table
logzilla events stats --ts-from "last 24 hours" --full-table
```

```
# Exclude archive or live data from the result
logzilla events stats --ts-from "3 days ago" --exclude-archive
```

Parser Performance

```
# View parser performance metrics
logzilla events parser-stats
```

Field Analysis

```
# Show field and tag cardinality (unique values)
logzilla events cardinality --scope all

# Filter fields by regex and compute exact values
logzilla events cardinality --scope fields --filter '^host|program$' --exact

# Show actual values seen (fields or tags)
logzilla events values --scope fields --limit 50
logzilla events values --scope tags --limit 100
```

Data Deletion

Permanently remove event data from LogZilla. Use these commands with extreme caution as data deletion cannot be undone.

```
# Permanently delete events for date range (timestamps must align to full hours)
logzilla drop --ts-from "2023-01-01 00:00" --ts-to "2023-01-31 00:00"

# Automatically round to the nearest hour
logzilla drop --ts-from "2023-01-01 00:05" --ts-to "2023-01-31 00:02" --round

# Skip confirmation (use with extreme caution)
logzilla drop --ts-from "2023-01-01 00:00" --ts-to "2023-01-31 00:00" --force-removal
```

Warning: The `drop` command permanently deletes data and cannot be undone. Always verify date ranges carefully. Use `--round` to correct non-hour-aligned timestamps.

Data Ingestion and Forwarding

Event Forwarding

Configure LogZilla to forward events to external systems such as SIEMs, notification platforms, or other log management systems.

For configuration structure, options, and examples, see the Forwarding Module (https://www.logzilla.ai/docs/forwarding-module/).

View Forwarder Configuration

```
# Display current forwarder configuration
logzilla forwarder print

# Display configuration split per files
logzilla forwarder print-files
```

Manage Forwarder Rules

```
# Import forwarder rules from file
logzilla forwarder import --input-file /path/to/forwarder.yaml

# Cleanup all forwarder configuration and counters (prompts for consent)
logzilla forwarder cleanup
```

Control Forwarder

```
# Reload forwarder configuration after changes
logzilla forwarder reload
```

Forwarder Statistics

```
# View forwarding statistics
logzilla forwarder stats

# View statistics for specific time range
logzilla forwarder stats --ts-from "2024-01-01 00:00" --ts-to "2024-01-31 23:59"
```

Network Traffic Analysis

Capture and analyze syslog traffic for troubleshooting ingestion issues and testing configurations.

Traffic Capture

```
# Start capturing syslog traffic
logzilla sniffer start --filename sniff-{t}.out --time 60 --live

# Store raw (unparsed) events
logzilla sniffer start --raw --time 120
```

Traffic Analysis

```
# View capture status
logzilla sniffer status

# Stop traffic capture
logzilla sniffer stop
```

Traffic Replay

```
# Replay captured traffic for testing
logzilla sniffer replay sniff-2025-01-01.out

# Keep original timestamps if present
logzilla sniffer replay sniff-2025-01-01.out --keep-ts
```

Event Processing Rules

Parser Rules Management

Manage rules that extract fields, normalize log messages, and enhance event data with additional context.

List and View Rules

```
# List all parser rules
logzilla rules list

# Filter by name using wildcards (quote the pattern)
logzilla rules list "*cisco*"

# Show recent runtime errors for rules
logzilla rules errors
```

Manage Rules

```
# Enable specific rule
logzilla rules enable "Custom Rule"

# Disable specific rule
logzilla rules disable "Old Rule"

# Add new rule from file
logzilla rules add /path/to/rule.yaml --name "Custom Rule"

# Remove rule
logzilla rules remove "Unused Rule"
```

Rule Testing and Validation

```
# Test rule for syntax errors
logzilla rules test "Custom Rule"

# Test rule against sample data
logzilla rules test --path /path/to/custom-rule.lua

# Validate rule file structure before adding
logzilla rules validate /path/to/new-rule.lua
```

```
# Dry run rule changes
logzilla rules test --path /path/to/custom-rule.lua --exitfirst
```

Rule Operations

```
# Reload all rules after changes
logzilla rules reload

# Export rules to file
logzilla rules export --output /backup/parser-rules.yaml

# Import rules from file
logzilla rules import --input /backup/parser-rules.yaml

# Run rule performance tests
logzilla rules performance --detailed
```

Trigger Management

Configure automated actions that execute when specific events occur, such as sending notifications, executing scripts, or creating tickets.

List and View Triggers

```
# List all triggers
logzilla triggers list

# Filter by name using wildcards (quote the pattern)
logzilla triggers list "*security*"
```

Manage Triggers

```
# Import triggers from file
logzilla triggers import --input /path/to/triggers.yaml --owner admin

# Export trigger to file
logzilla triggers export --trigger-id 123 --output /backup/security-trigger.yaml

# Export all triggers
logzilla triggers export --output /backup/all-triggers.yaml

# Update trigger configuration fields
logzilla triggers update --trigger-id 123 --set add_note=true severity=4
```

Delete Triggers

```
# Delete by id or by name filter (prompts for consent)
logzilla triggers delete --trigger-id 123
logzilla triggers delete "*old*"
```

Trigger Performance

```
# Run trigger performance tests
logzilla triggers performance
```

Data Quality and Maintenance

Repair and Cleanup

```
# Run automated repair for storage and aggregates
logzilla events fix --ts-from "2024-01-01 00:00" --ts-to "2024-01-02 00:00" --scope storage
logzilla events fix --ts-from "2024-01-01 00:00" --ts-to "2024-01-02 00:00" --scope aggregates
```

Notes

- Some operations require LogZilla to be running and properly licensed.
- Timestamps that represent date ranges should align to full hours for storage operations. Use `--round` where available.

Best Practices

Data Lifecycle Management

- **Implement retention policies** based on compliance requirements
- **Archive old data regularly** to manage disk space
- **Monitor storage usage** and plan for growth
- **Test archive and restore procedures** regularly

Optimizing Performance

- **Monitor parser performance** and optimize rules
- **Use appropriate archival schedules** for your data volume
- **Regular maintenance tasks** to keep system optimized
- **Monitor forwarder performance** and adjust as needed

Data Quality

- **Validate parser rules** before deployment

- **Monitor trigger execution** for errors
- **Regular data integrity checks** to ensure consistency
- **Test configurations** in development environments

Security and Compliance

- **Secure forwarder connections** with encryption
- **Audit data access** and modification activities
- **Implement proper data retention** for compliance
- **Monitor and log administrative actions**

These data management commands provide control over the LogZilla data lifecycle. Proper use of these tools ensures optimal performance, compliance with retention policies, and reliable data processing workflows.