LDP Deployment and Operations

Wayne Schneider <wayne@indexdata.com>
Index Data <https://www.indexdata.com>

FOLIO SysOps SIG
29 April 2022
Simple schematic (LDP 1.X)

FOLIO with microservices

okapi

Scheduling service

LDP server

ldpmarc

folio-analytics (SQL scripts)

Reporting DB

Operational DB
LDP reporting DB

LDP Admin Guide:

Highlights:

- Sizing: 4 CPU cores, 32 G RAM, 1 TB storage as base, adjust for smaller/larger datasets
- PostgreSQL 13.6+, 14.2+ recommended
- Recommended configuration settings
- One database per FOLIO tenant
LDP reporting DB schemata

- **public**: contains LDP reporting data extracted from operational DB and FOLIO
- **history**: contains versioned history of data from public tables
- **local**: user workspace for creation of supplemental data or local derived data
- **folio_reporting**: derived tables from the folio-analytics community project
- **dbconfig**: configuration options for LDP server behavior
- **dbsystem**: system administrative data, e.g. server logs, foreign keys, etc.
LDP reporting DB users

Required users:

- ldpadmin: owns all reporting DB tables and schemata
- ldpconfig: user account for making changes in the dbconfig schema
- ldp: user account with read permissions for reporting data in public, history, and folio_reporting schemata, read/write permissions for local schemata

User accounts:

- Manage user account permissions with LDP
LDP server build and deploy

Scheduling service for daily build:

- cron
- Job scheduling server, e.g. Jenkins
- Kubernetes cronjob

Requirements:

- Linux (Debian recommended)
- Network proximity to FOLIO, operational DB, reporting DB
- 1 CPU core/1 GB RAM per LDP server instance running concurrently

Deployment options:

- Build from source, run executable
- Run from container in Github container registry
LDP server configuration

```json
{
  "deployment_environment": "production",
  "ldp_database": {
    "database_name": "ldp",
    "database_host": "ldp.folio.org",
    "database_port": 5432,
    "database_user": "ldpadmin",
    "database_password": "(ldpadmin password here)",
    "database_super_user": "postgres",
    "database_super_password": "(postgres password here)",
    "database_sslmode": "require"
  },
  "enable_sources": ["my_library"],
  "sources": {
    "my_library": {
      "okapi_tenant": "diku",
      "direct_database_name": "diku_prod",
      "direct_database_host": "database.folio.org",
      "direct_database_port": 5432,
      "direct_database_user": "ldp",
      "direct_database_password": "(database password here)"
    }
  }
}
```
ldpmarc

Deployment highlights:

● Requires ldp 1.6+
● Creates srs_marctab table in public schema of LDP reporting DB
● Significant storage requirement – additional 500 GB in reporting DB
● Must enable pg_tgrm extension with CREATE EXTENSION
● Uses same configuration file as LDP server
● After initial build, daily incremental update is quite fast (10 minutes for UChicago)
folio_reporting tables (folio-analytics derived tables)

- SQL scripts in folio-analytics sql/derived_tables
  - Community-contributed
  - Versioned and released to work with specific LDP versions

Requirements:

- psql to run the scripts
- Control shell script to read runlist.txt and execute scripts in order
mod-ldp and the LDP query builder app (Kiwi+)

mod-ldp:

- FOLIO backend module that communicates with the LDP reporting DB
- Proxied by Okapi
- Uses FOLIO operational DB to store configuration and secrets

LDP query builder:

- Stripes module that can be included in a FOLIO/Stripes webpack
- Uses the API provided by mod-ldp to run simple reporting queries and return results in the FOLIO UI
mod-ldp configuration

POST reporting DB credentials to /ldp/config/dbinfo:

```json
{
  "key": "dbinfo",
  "tenant": "diku",
  "value": "{" "user": \"ldp_user\",
                "url": \"jdbc:postgresql://somehost.net:5432\/db_name\",
                "pass": \"abc123def456\"
    }"
}
```