LOCKSS System Manual

LOCKSS Program

Nov 22, 2022

LOCKSS 2.0-ALPHA2 SYSTEM MANUAL

Latest release: 2.0.26-alpha2 (2020-02-26)
First release: 2.0.21-alpha2 (2020-02-06)
System manual last built: Nov 22, 2022

Welcome to the LOCKSS 2.0-alpha2 System Manual.

CHAPTER

ONE

INSTALLING THE LOCKSS SYSTEM

This section describes how to install the LOCKSS system.

1.1 System Prerequisites

1.1.1 Machine

The LOCKSS system runs on a **64-bit Linux** host (physical or virtual), with **4 cores** (8 or more preferable) and **8 GB of memory** (16 GB or more preferable).

1.1.2 Operating System

The LOCKSS system requires a **64-bit Linux** host compatible with Systemd and Docker **18.09 or better**.

Many Linux distributions have systemd and can run Docker 18.09 or better. To name a few that are commonly used with the LOCKSS system:

- Arch Linux
- CentOS 7
- Debian 9 (Stretch)
- Fedora 28 or better
- Oracle Linux 7
- Ubuntu 16.04 LTS (Xenial) or better

1.1.3 User

The LOCKSS system runs under a system user named `lockss` under a group named `lockss`, which you will need to create. **For alpha2 only, the lockss user must have sudo privileges.**

To check installed prerequisites, use the script *check-sys*, which will attempt to find and install missing elements.

1.2 Installing Docker

Docker is a container runtime and orchestration engine. This page will walk you through the initial installation of the Docker engine and the Local-Persist Docker volume plugin.

1.2.1 Overview

To install Docker for the purposes of running the LOCKSS system:

- Install Docker
- Check the System Group
- Start Docker
- Enable Docker at Startup
- Check the Storage Driver
- Initialize Swarm Mode
- Install Local-Persist
- Reconfiguring Docker (if required along the way)

1.2.2 Install the Docker Engine

The LOCKSS system requires **Docker 18.09 or better**.

Docker on Arch Linux

Simply use Arch's official software repositories to install Docker with Pacman:

```
sudo pacman -S docker
```

Docker on CentOS

> **Caution:** CentOS 7 is required.

Use Docker's official software repositories to install Docker with Yum: https://docs.docker.com/install/linux/docker-ce/centos/ (the version of Docker available through the standard CentOS software repositories is not suitably recent).

Docker on Debian

Caution: Debian 9 (Stretch) is required.

Use Docker's official software repositories to install Docker with Apt: https://docs.docker.com/install/linux/docker-ce/debian/ (the docker package available through the standard Debian software repositories is for an unrelated system tray application).

Docker on Fedora

Caution: Fedora 28 or better is required.

Use Docker's official software repositories to install Docker with Dnf: https://docs.docker.com/install/linux/docker-ce/fedora/ (the version of Docker available through the standard Fedora software repositories is not suitably recent).

Docker on Oracle Linux

Caution: Oracle Linux 7 is required.

Use Oracle's official software repositories to install Docker with Yum: https://docs.oracle.com/cd/E52668_01/E87205/html/section_install_upgrade_yum_docker.html.

Docker on Ubuntu

Caution: Ubuntu 16.04 LTS (Xenial) or better is required.

Use Docker's official software repositories to install Docker with Apt: https://docs.docker.com/install/linux/docker-ce/ubuntu/ (the docker package available via the Ubuntu Universe software repository is for an unrelated desktop system tray application).

1.2.3 Check the System Group

Installing Docker creates a new system group typically named `docker`. The command:

```
groups lockss
```

will display the list of groups the `lockss` user is a member of, which should already include the `lockss` group. If the list of groups does not include the `docker` group, add the `lockss` user to the `docker` group with the following command:

```
sudo usermod -G docker -a lockss
```

or with a similar user management command or tool.

1.2.4 Start Docker

Start Docker with `systemd`:

```
sudo systemctl start docker
```

Verify that Docker is running:

```
sudo systemctl is-active docker
```

The output should say `active`.

1.2.5 Enable Docker at Startup

Unless you are only trying out the LOCKSS system on a machine that will not be running it or Docker routinely, enable Docker to launch at startup with `systemd`:

```
sudo systemctl enable docker
```

Verify that the operation succeeded with:

```
sudo systemctl is-enabled docker
```

The output should say `enabled`.

1.2.6 Check the Storage Driver

Verify that Docker is using the OverlayFS (`overlay2`) storage driver:

```
sudo -u lockss docker info | grep 'Storage Driver:'
```

If the output is:

```
Storage Driver: overlay2
```

then Docker is running with the OverlayFS storage driver and you can move on to the next section.

If the output lists another storage driver than `overlay2` (for example `devicemapper`), see the Reconfiguring Docker section below, add the key-value pair `"storage-driver":"overlay2"` to `/etc/docker/daemon.json`, and restart the Docker daemon.

1.2.7 Initialize Swarm Mode

Verify that Docker is using Swarm mode:

```
sudo -u lockss docker info | grep 'Swarm:'
```

If the output is:

```
Swarm: active
```

then Docker is running in Swarm mode correctly and you can move on to the next section.

If the output is empty or if the Swarm is not listed as active, initialize Swarm mode with this command:

```
sudo -u lockss docker swarm init
```

If the output looks like this:

```
Swarm initialized: current node (bvz81updecsj6wjz393c09vti) is now a manager.
To add a worker to this swarm, run the following command:
    docker swarm join \
    --token SWMTKN-1-3pu6hszjas19xyp7ghgosyx9k8atbfcr8p2is99znpy26u2lkl-
↪1awxwuwd3z9j1z3puu7rcgdbx \
    xx.xx.xx.xx:2377
To add a manager to this swarm, run 'docker swarm join-token manager' and follow the␣
↪instructions.
```

where `xx.xx.xx.xx` is the IP address of the machine, then the Swarm initialization was successful and you can move on to the next section.

If the output contains an error message that looks like this:

```
Error response from daemon: could not choose an IP address to advertise since this␣
↪system has multiple addresses on interface eth0 (xx.xx.xx.xx and yy.yy.yy.yy) -␣
↪specify one with --advertise-addr
```

or like this:

```
Error response from daemon: could not choose an IP address to advertise since this␣
↪system has multiple addresses on different interfaces (xx.xx.xx.xx on eth0 and yy.yy.
↪yy.yy on eth1) - specify one with --advertise-addr
```

then Docker was not able to automatically select an IP address among the several IP addresses it discovered. Identify the desired IP address of the machine, for example `xx.xx.xx.xx`, and enter the modified command:

```
sudo -u lockss docker swarm init --advertise-addr xx.xx.xx.xx
```

1.2.8 Install Local-Persist

The LOCKSS system uses the Local-Persist Docker volume plugin.

Run the Local-Persist installation script which will download and install the necessary files and set up the necessary `systemd` infrastructure:

```
curl -fsSL https://raw.githubusercontent.com/MatchbookLab/local-persist/master/scripts/
↪install.sh | sudo bash
```

If you need to use Upstart instead of Systemd:

```
curl -fsSL https://raw.githubusercontent.com/MatchbookLab/local-persist/master/scripts/
↪install.sh | sudo bash -s -- --upstart
```

Verify that Local-Persist is running and enabled at startup:

```
sudo systemctl is-active docker-volume-local-persist
```

should say `active`; and

```
sudo systemctl is-enabled docker-volume-local-persist
```

should say `enabled`.

Finally verify that Local-Persist is ready for use by creating a temporary volume. To create a volume named `foo` which stores files in the directory `/tmp/foo`:

```
docker volume create --name=foo -d local-persist -o mountpoint=/tmp/foo

docker volume inspect foo
```

The output should look something like this:

```
[
    {
        "CreatedAt": "0001-01-01T00:00:00Z",
        "Driver": "local-persist",
        "Labels": {},
        "Mountpoint": "/tmp/foo",
        "Name": "foo",
        "Options": {
            "mountpoint": "/tmp/foo"
        },
        "Scope": "local"
    }
]
```

Once you verify the mountpoint, you can delete it:

```
docker volume rm foo

rm -rf /tmp/foo
```

1.2.9 Reconfiguring Docker

This section describes what to do when Docker needs to be reconfigured. You do not need to do anything unless one of the sections above sends you here.

Edit or create the `/etc/docker/daemon.json` configuration file and input the required key-value pairs in a JSON object, separated by commas, typically one per line for clarity. Example:

```
{
    "storage-driver": "overlay2",
    "iptables": true
}
```

After editing and saving the configuration file, restart the Docker daemon with systemd:

```
sudo systemctl restart docker
```

1.3 Installing Git

Git is a version control system, used to interact with code repositories.

The LOCKSS Installer is available from GitHub, and you will need a Git client to download it.

Your operating system may already be equipped with a Git client. Type:

```
git --version
```

If the output is a version number, for example:

```
git version 2.21.0
```

then Git is already installed and you do not need to take any further action.

Otherwise, Git can be installed from your operating system's software repositories.

1.3.1 Git on Arch Linux

Use Pacman to install Git on Arch Linux:

```
sudo pacman -S git
```

1.3.2 Git on CentOS

Caution: CentOS 7 is required.

Use Yum to install Git on CentOS:

```
sudo yum install git
```

1.3.3 Git on Debian

Caution: Debian 9 (Stretch) is required.

Use Apt to install Git on Debian:

```
sudo apt-get install git
```

1.3.4 Git on Fedora

Caution: Fedora 28 or better is required.

Use Dnf to install Git on Fedora:

```
sudo dnf install git
```

1.3.5 Git on Oracle Linux

Caution: Oracle Linux 7 is required.

Use Yum to install Git on Oracle Linux:

```
sudo yum install git
```

1.3.6 Git on Ubuntu

Caution: Ubuntu 16.04 LTS (Xenial) or better is required.

Use Apt to install Git on Ubuntu:

```
sudo apt-get install git
```

1.4 Downloading the LOCKSS Installer

You can download the LOCKSS Installer from GitHub using a *Git* command as the `lockss` user created in *System Prerequisites*:

```
git clone https://github.com/lockss/lockss-installer
```

You can then enter the `lockss-installer` directory:

```
cd lockss-installer
```

1.5 Checking the System

After *installing the LOCKSS system* and *downloading the LOCKSS Installer*, prepare the system for running by typing:

```
scripts/check_sys
```

in the `lockss-installer` directory.

The script will do its best to install any missing elements needed to run the LOCKSS cluster on the host machine. See the *System Prerequisites* document for required system elements.

1. Check for Docker and install if missing.
2. Check for the Local-Persist Docker volume plugin and install if missing.
3. Ensure Docker Swarm is initialized and running.
4. Check for a user `lockss` and create the `lockss` user and group if missing.

CHAPTER

TWO

UPGRADING FROM LOCKSS 2.0-ALPHA1

If you have been using version 2.0-alpha1 of the LOCKSS system, an upgrade path has been provided to version 2.0-alpha2.

2.1 Prerequisites

See *Installing the LOCKSS System*. Additionally you will need to install OpenJDK8 on the host machine.

Note: This dependency on Java is temporary for 2.0-alpha2. It is necessary to run the Solr upgrader tool. In 2.0-alpha3, the process will be packaged in such a way that it does not depend on Java on the host machine.

2.2 Update `lockss-installer`

On the command line in the `lockss-installer` directory, type:

```
git checkout master

git pull
```

to update to the latest version of `lockss-installer` from GitHub.

2.3 Run the Upgrade Command

On the command line in the `lockss-installer` directory, type:

```
sudo scripts/upgrade-alpha1-to-alpha2
```

The script will perform a number of system-level changes and need to be `root`.

These adjustments include renaming `config.info` to `system.cfg`, shutting down a running system stack, renaming storage directories, updating database names, run the Solr upgrader tool from 2.0-alpha1 to 2.0-alpha2 (which is a long running process; please be patient), and change file ownerships, all of which to align the system with the 2.0-alpha2 environment.

2.4 Re-Configure the System

Upon successful completion, you will prompted to run `scripts/configure-lockss`. **Be advised that the configuration process will prompt you for the Postgres database password.**

CHAPTER

THREE

CONFIGURING THE LOCKSS SYSTEM

After *Installing the LOCKSS System* and *Downloading the LOCKSS Installer*, configure the system with the configure script:

```
scripts/configure-lockss
```

(If you have experience with classic LOCKSS daemon version 1.x, this is the equivalent of `hostconfig`.)

When run the first time the questions asked by the script often come with a suggested value, displayed in square brackets; hit `Enter` to accept the suggested value, or type the correct value and hit `Enter`. Any subsequent runs will use the previous values ad the suggested value, review and hit `Enter` to leave unchanged.

Questions include:

1. `Fully qualified hostname (FQDN) of this machine:` Enter the machine's hostname (e.g. `lockstest.myuniversity.edu`).
2. `IP address of this machine:` The publicly routable IP address of the machine, or if it is not publicly routable but will be accessible via network address translation (NAT), its IP address on the internal network.
3. `Is this machine behind NAT?:` Enter `Y` if the machine is not publicly routable but will be accessible via network address translation (NAT), or `N` otherwise.
    1. `External IP address for NAT:` If you answered `Y` to the previous question, enter the publicly routable IP address of the machine.
4. `Initial subnet for admin UI access:` Enter a semicolon-separated list of subnets in CIDR notation that should have access to the Web user interfaces of the system.
5. `LCAP V3 protocol port:` Enter the port on the publicly routable IP address that will be used to receive LCAP (LOCKSS polling and repair) traffic. Historically, most LOCKSS nodes use 9729.
6. `PROXY port:` Not yet re-enabled in 2.0-alpha; ignore.
7. `Mail relay for this machine:` Hostname for this machine's outgoing mail server.
8. `Does mail relay <mailhost> need user & password`: Enter `Y` if the outgoing mail server requires password authentication, `N` otherwise.
    1. `User for <mailhost>:` If you answered `Y` to the outgoing mail server password authentication question, enter the username for the mail server.
    2. `Password for <mailuser>@<mailhost>:` Enter the password for the given username.
    3. `Again:` Re-enter the mail server password (if the two passwords do not match, the password will be asked again).
9. `E-mail address for administrator:` Enter the e-mail address of the person or team who will administer the LOCKSS system on this machine.

10. `Configuration URL:` Enter the URL of the LOCKSS network configuration file. If you are not running your own LOCKSS network, use `http://props.lockss.org:8001/demo/lockss.xml`, the configuration file for a demo network set up for LOCKSS 2.0 pre-release testing.
11. `Configuration proxy (host:port):` enter a host:port combination for the proxy server needed to reach the network configuration file (or simply hit Enter if none is needed).
12. `Preservation group(s):` Enter a semicolon-separated list of preservation network identifiers. If you are not joining an existing network or running your own, enter demo, the network identifier for the demo network set up for LOCKSS 2.0 pre-release testing.
13. `Content data storage directory:` Enter the path of a directory that is the root of the main storage area of the LOCKSS system. If you are used to the classic LOCKSS daemon (1.x), this would be the equivalent of `/cache0`.
14. `Service logs directory:` Enter the path of a directory that is the root of the storage area for LOCKSS-related log files (historically `/var/log/lockss`).
15. `Temporary storage directory:` not actively used in LOCKSS 2.0-alpha2; ignore.
16. `User name for web UI administration:` Enter the username for an administrative user in the LOCKSS system's Web user interfaces.
17. `Password for web UI administration user <uiuser>:` Enter the password for the given administrative user in the LOCKSS system's Web user interfaces[1].
18. `Password for web UI administration user <uiuser> (again):` Re-enter the password for the given administrative user in the LOCKSS system's Web user interfaces (if the two passwords do not match, the password will be asked again).

The next set of questions will gather information about which of the LOCKSS services you will be using and how to access any service you have already configured for use:

19. `Use LOCKSS Metadata Query Service?:` Enter `Y` to use the included metadata service or `N` and no metadata service will be run.
20. `Use LOCKSS Metadata Extractor Service?:` Enter `Y` to use the included metadata extraction service or `N` and no metadata service will be run.
21. `Use LOCKSS PostgreSQL DB Service?`:
    - Enter `Y` to use the embedded PostgreSQL database.
        1. `Password for database:` Enter the password for the PostgreSQL database included in LOCKSS 2.0-alpha2[1].
        2. `Password for database (again):` Re-enter the password for the PostgreSQL database (if the two passwords do not match, the password will be asked again).
    - Enter `N` if you wish to use your own PostgreSQL database. You will be queried for the details of your PostgreSQL service.
        1. `Fully qualified hostname (FQDN) of PostgreSQL host:` Enter the hostname of your PostgreSQL database (e.g. `mypgsql.myuniversity.edu`).
        2. `Port used by PostgreSQL host:` Enter the port where your running PostgreSQL database can be reached.
        3. `Login name for PostgreSQL service:` Enter the user name for your PostgreSQL database. The default is `LOCKSS`.

[1] Passwords are encrypted in the Docker Secret vault. You should also keep your passwords in a safe place for yourself, as you will need them each time you run `scripts/configure-lockss`. If you change your password in PostgreSQL, you will need to re-run `scripts/configure-lockss` to give the new password to the system.

4. `Schema for PostgreSQL service:` Enter the schema name to be used by the LOCKSS system. The default is `LOCKSS`.
5. `Database name prefix for PostgreSQL service:` Prefix to use for any LOCKSS databases. The default is `Lockss` (note the uppercase/lowercase).
6. `Password for PostgreSQL database:` Enter the password for your PostgreSQL database[1].
7. `Password for PostgreSQL database (again):` Re-enter the password for your PostgreSQL database (if the two passwords do not match, the password will be asked again).

22. `Use LOCKSS Solr Service?:`

- Enter `Y` if you wish to use the included Solr install.
- Enter `N` if you wish to use your own Solr database.

1. `Fully qualified hostname (FQDN) of Solr host:` Enter the hostname of your Solr database server (e.g. `mysolr.myuniversity.edu`).
2. `Port used by Solr host:` Enter the port where your running Solr database server can be reached.
3. `Solr core repo name:` Enter name of the Solr core for the LOCKSS repository. The default is `lockss-repo`.

23. `Use LOCKSS PyWb Service?:` Answer `Y` to use PyWb, answer `N` and you will be offered the option to use OpenWayback.
24. `OK to store this configuration:` Confirm with `Y` that the summarized configuration data is correct and that you are ready to write it to a file.

You will prompted to run `scripts/start-lockss` to start the configured system.

CHAPTER

FOUR

RUNNING THE LOCKSS SYSTEM

After *Configuring the LOCKSS System* and anytime after updating the LOCKSS Installer to a new version and stopping your LOCKSS stack:

- Starting the LOCKSS system

  Run `scripts/start-lockss`: This script will calls in turn:

  - `scripts/generate-lockss`. This script takes your configuration data and turns into into a set of configuration files containing the right values.
  - `scripts/assemble-lockss`. This script puts the configuration files and puts them in the right places, and ensures that all storage volumes are ready for use (creating them if necessary).
  - `scripts/deploy-lockss`. This script deploys your LOCKSS stack by invoking Docker.

- Shutting down the LOCKSS system

  `scripts/shutdown-lockss`

- To restart a running or shut down LOCKSS 2.0-alpha2 cluster:

  `scripts/restart-lockss`

- To remove all configurations, volumes and networks installed by LOCKSS from Docker.

  Run `scripts/uninstall-lockss`: Remove all lockss elements from docker. This will not remove files from the persistent store.

CHAPTER

FIVE

USING THE LOCKSS SYSTEM

This section describes how to use the LOCKSS system.

5.1 Using the LOCKSS Configuration Service

Note: This page is under construction.

5.1.1 Accessing the Web User Interface

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *Config Service* in the top-left menu.

Alternatively, if your primary hostname is *<HOST>*, you can use your browser to connect to the LOCKSS Configuration Service Web user interface (UI) at `http://<HOST>:24621`.

Enter your Web UI username and password to login if prompted.

5.1.2 Adding Archival Units

To add AUs to the system for preservation:

1. In the top-right menu, click *Journal Configuration*.
2. In the center menu, click *Add AUs*.
3. Select one or more collections of AUs by selecting the checkbox next to the appropriate collection.
4. Click the *Select AUs* button. It may take a bit of time (60+ seconds) for the next screen to appear, while the list of AUs is built.
5. Select one or more AUs from the AU list. You may click *Select All* if you would like to select all AUs. If you choose to use select all AUs, please note that the next step may take some time to load.
6. Click the *Add Selected AUs* button. The time it takes for the page to refresh depends on the number of AUs added. Give the LOCKSS system some time to load the AUs and reload the page before moving on.
7. A screen will show a list of added AUs. Crawling of these new AUs will start automatically – no further action is necessary unless prompted by a footnote next to an AU's name.

5.1.3 Configuring a Crawl Proxy

If Web crawls must be routed through a Web proxy:

1. In the top-right menu, click *Content Access Options*.
2. In the center menu, click *Proxy Client Options*.
3. Select the *Proxy crawls* checkbox.
4. Enter the hostname and port of the Web proxy in the *HTTP Proxy host* and *Port* text areas, respectively.
5. Click the *Update Proxy Client* button.

5.1.4 Managing Access to the Web User Interfaces

This section is under construction.

5.2 Using the LOCKSS Metadata Extraction Service

Note: This page is under construction.

5.2.1 Accessing the Web User Interface

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *Metadata Extraction Service* in the top-left menu.

Alternatively, if your primary hostname is *<HOST>*, you can use your browser to connect to the LOCKSS Configuration Service Web user interface (UI) at `http://<HOST>:24641`.

Enter your Web UI username and password to login if prompted.

5.2.2 Requesting Metadata Extraction

This section is under construction.

5.3 Using the LOCKSS Metadata Service

Note: This page is under construction.

5.3.1 Accessing the Web User Interface

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *Metadata Service* in the top-left menu.

Alternatively, if your primary hostname is *<HOST>*, you can use your browser to connect to the LOCKSS Configuration Service Web user interface (UI) at `http://<HOST>:24651`.

Enter your Web UI username and password to login if prompted.

5.3.2 Requesting Metadata Information

This section is under construction.

5.4 Using the LOCKSS Poller Service

Note: This page is under construction.

5.4.1 Accessing the Web User Interface

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *Poller Service* in the top-left menu.

Alternatively, if your primary hostname is *<HOST>*, you can use your browser to connect to the LOCKSS Configuration Service Web user interface (UI) at `http://<HOST>:24631`.

Enter your Web UI username and password to login if prompted.

5.4.2 Requesting Polls

This section is under construction.

5.4.3 Monitoring Polling and Voting

This section is under construction.

5.5 Using the LOCKSS Crawler Service

Note: This page is under construction.

5.5.1 Accessing the Web User Interface

Note: Currently the crawler service is run as part of the poller service.

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *Crawler Service* in the top-left menu.

Alternatively, if your primary hostname is *<HOST>*, you can use your browser to connect to the LOCKSS Configuration Service Web user interface (UI) at `http://<HOST>:24631`.

Enter your Web UI username and password to login if prompted.

5.5.2 Monitoring Crawl Status in the System

The Crawl status of all configured AUs is available in the Archival Unit table

1. In the top-right menu, click *Daemon Status*.
2. Open the control in the middle of the screen that says *Overview* and select *Archival Units:guilabel:* from the drop down menu.
    - If prompted, enter your Username and Password again.
    - It will take a bit of time for the next screen to appear while the AU list is being built.
3. The Archival Units screen lists statistics for each configured AU

- the *Last Successful Crawl* column provides a timestamp of the most recent sucessful crawl.
- the *Last Crawl Start* column provides a timestamp of the last attempted crawl.
- the *Last Crawl Result* column provides the exit status of the last attempted crawl.

5.5.3 Causing an Archival Unit to Crawl

Archival units (AUs) that have been added to the system for preservation crawl periodically, but you can cause an AU to crawl on demand:

1. In the top-right menu, click *Debug Panel*.
2. Select an AU in the *AU Actions: select AU* drop-down list.
3. Click the *Start Crawl* button.
4. If the AU has crawled recently, you will be prompted to confirm that you wish to override the usual recrawl interval by clicking on the *Force Start Crawl* button.

5.5.4 Crawl Status Screen

To inspect the state of crawls, access the *Crawl Status* screen:

1. In the top-right menu, click *Daemon Status*.
2. In the center drop-down list, select *Crawl Status*. Alternatively, in the center overview, click on the second line, which says "*N* active crawls".

Top-Level Crawl Information

The top left of the Crawl Status table contains the number of active, successful or failed crawls, and a countdown until the next time the system will look at the AUs being preserved and pick some that are ready to crawl or recrawl.

Crawl Status Entry

Each line in the Crawl Status table contains:

- The name of the AU
- The type of crawl
- The start time of the crawl
- The duration of a finished or in-progress crawl
- The status of the crawl
- The number of bytes fetched over the network as part of the crawl
- The number of URLs fetched as part of the crawl
- The number of URLs parsed for more links
- The number of URLs remaining to be fetched as part of this crawl
- The number of URLs encountered as part of this crawl but excluded from being fetched
- The number of URLs fetched as part of the crawl, that received an HTTP Not Modified response
- The number of URLs that caused errors as part of this crawl
- The number of different content types encountered as part of the crawl

Most of these values can be clicked to see a list of the corresponding objects.

5.6 Replaying Web Content with Pywb

5.6.1 Accessing the Pywb User Interface

Given that your primary hostname is samp:*{<HOST>}*, you can use your browser to connect to the Pywb user interface (UI) at `http://<HOST>:8080`.

5.6.2 Replaying a URL

To view a URL from Pywb:

1. The Pywb screen provides a list of links to available collections. Click on the top-most collection which should be `/lockss`.
2. Enter the URL you want to replay in the URL search box.
3. Click the *Search* button.
4. Replay the most recent URL by clicking on the topmost entry of the third column.

5.6.3 Finding a URL From an AU to Replay

There are multiple ways to discover URLs belonging to an AU in the Configuration Service UI:

1. Obtaining a URL by clicking on "pages fetched" inside of crawl status
    - In the top-right menu, click *Daemon Status*.
    - Open the control in the middle of the screen that says *Overview* and select *Crawl Status* from the drop down menu.
    - Picking an AU from the active crawls, click on the number associated with *Pages Fetched* to bring up a list of URLs that have been crawled.
    - Copy one of the URLs and paste it in the Pywb interface as described previously.
2. Obtaining a Substance URL
    - In the top-right menu, click *Daemon Status*.
    - Open the control in the middle of the screen that says *Overview* and select *Archival Units* from the drop down menu. If prompted, enter your Username and Password again. It will take a bit of time for the next screen to appear while the AU list is being built.
    - Select an AU by clicking on the AU title in the first column.
    - Open the *Substance URLs* link
    - Copy one of the URLs and paste it in the Pywb interface as described previously.

5.7 Replaying Web Content with OpenWayback

5.7.1 Accessing the OpenWayback User Interface

Given that your primary hostname is samp:*{<HOST>}*, you can use your browser to connect to the Pywb user interface (UI) at `http://<HOST>:8080/wayback`.

5.7.2 Replaying a URL

To view a URL from OpenWayback:

1. Enter the URL you want to replay in the URL search box.
2. Click the *Search* button.
3. Select the *Year* or leave as :guilabel:`All*
4. Click *Take Me Back*.

5.7.3 Finding a URL From an AU to Replay

There are multiple ways to discover URLs belonging to an AU in the Configuration Service UI:

1. Obtaining a URL by clicking on "pages fetched" inside of crawl status
    - In the top-right menu, click *Daemon Status*.
    - Open the control in the middle of the screen that says *Overview* and select *Crawl Status* from the drop down menu.
    - Picking an AU from the active crawls, click on the number associated with *Pages Fetched* to bring up a list of URLs that have been crawled.
    - Copy one of the URLs and paste it in the OpenWayback interface as described previously.
2. Obtaining a Substance URL
    - In the top-right menu, click *Daemon Status*.
    - Open the control in the middle of the screen that says *Overview* and select *Archival Units* from the drop down menu. If prompted, enter your Username and Password again. It will take a bit of time for the next screen to appear while the AU list is being built.
    - Select an AU by clicking on the AU title in the first column.
    - Open the *Substance URLs* link
    - Copy one of the URLs and paste it in the OpenWayback interface as described previously.

CHAPTER

SIX

APPENDIX

This appendix contains additional pages of information about the LOCKSS system.

6.1 Release Notes

6.1.1 LOCKSS 2.0.26-alpha2

Released: 2020-02-26
Also known as: LOCKSS 2.0-alpha2f

LOCKSS 2.0.26-alpha2 (also known as LOCKSS 2.0-alpha2f) is a bug fix release and the latest release of the LOCKSS 2.0-alpha2 system. It addresses bugs in the LOCKSS Installer.

Release Notes

- Bug fixes in installer scripts.

Component Versions

LOCKSS 2.0.26-alpha2 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.26-alpha2
- LOCKSS Repository Service version 2.0.9.0
- LOCKSS Configuration Service version 2.0.3.0
- LOCKSS Poller Service version 2.0.1.0
- LOCKSS Metadata Extraction Service version 2.0.2.0
- LOCKSS Metadata Service version 2.0.1.0
- PostgreSQL version 9.6.12
- Apache Solr version 7.2.1
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

6.1.2 LOCKSS 2.0.25-alpha2

Released: 2020-02-25
Also known as: LOCKSS 2.0-alpha2e

LOCKSS 2.0.25-alpha2 (also known as LOCKSS 2.0-alpha2e) is a bug fix release of the LOCKSS 2.0-alpha2 system. It addresses bugs in the LOCKSS Installer.

Release Notes

- Bug fixes in installer scripts.
- Bug fixes in the included Solr.

Component Versions

LOCKSS 2.0.25-alpha2 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.25-alpha2
- LOCKSS Repository Service version 2.0.9.0
- LOCKSS Configuration Service version 2.0.3.0
- LOCKSS Poller Service version 2.0.1.0
- LOCKSS Metadata Extraction Service version 2.0.2.0
- LOCKSS Metadata Service version 2.0.1.0
- PostgreSQL version 9.6.12
- Apache Solr version 7.2.1
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

6.1.3 LOCKSS 2.0.24-alpha2

Released: 2020-02-25
Also known as: LOCKSS 2.0-alpha2d

LOCKSS 2.0.24-alpha2 (also known as LOCKSS 2.0-alpha2d) is a bug fix release of the LOCKSS 2.0-alpha2 system. It addresses bugs in the LOCKSS Installer.

Release Notes

- Bug fixes in installer scripts.
- Bug fixes in the included Solr.

Component Versions

LOCKSS 2.0.24-alpha2 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.24-alpha2
- LOCKSS Repository Service version 2.0.9.0
- LOCKSS Configuration Service version 2.0.3.0
- LOCKSS Poller Service version 2.0.1.0
- LOCKSS Metadata Extraction Service version 2.0.2.0
- LOCKSS Metadata Service version 2.0.1.0
- PostgreSQL version 9.6.12
- Apache Solr version 7.2.1
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

6.1.4 LOCKSS 2.0.23-alpha2

Released: 2020-02-16
Also known as: LOCKSS 2.0-alpha2c

LOCKSS 2.0.23-alpha2 (also known as LOCKSS 2.0-alpha2c) is a bug fix release of the LOCKSS 2.0-alpha2 system. It addresses bugs in the LOCKSS Installer.

Release Notes

- Bug fixes in installer scripts.
- Fix upgrade logic when the included Solr has never run.

Component Versions

LOCKSS 2.0.23-alpha2 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.23-alpha2
- LOCKSS Repository Service version 2.0.9.0
- LOCKSS Configuration Service version 2.0.3.0
- LOCKSS Poller Service version 2.0.1.0
- LOCKSS Metadata Extraction Service version 2.0.2.0
- LOCKSS Metadata Service version 2.0.1.0
- PostgreSQL version 9.6.12

- Apache Solr version 7.2.1
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

6.1.5 LOCKSS 2.0.22-alpha2

Released: 2020-02-10
Also known as: LOCKSS 2.0-alpha2b

LOCKSS 2.0.22-alpha2 (also known as LOCKSS 2.0-alpha2b) is a bug fix release of the LOCKSS 2.0-alpha2 system. It addresses bugs in the LOCKSS Installer.

Release Notes

- Run certain scripts only as the `lockss` user.

Component Versions

LOCKSS 2.0.22-alpha2 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.22-alpha2
- LOCKSS Repository Service version 2.0.9.0
- LOCKSS Configuration Service version 2.0.3.0
- LOCKSS Poller Service version 2.0.1.0
- LOCKSS Metadata Extraction Service version 2.0.2.0
- LOCKSS Metadata Service version 2.0.1.0
- PostgreSQL version 9.6.12
- Apache Solr version 7.2.1
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

6.1.6 LOCKSS 2.0.21-alpha2

Released: 2020-02-06
Also known as: LOCKSS 2.0-alpha2a

LOCKSS 2.0.21-alpha2 (also known as LOCKSS 2.0-alpha2a) is the first release of the LOCKSS 2.0-alpha2 system.

Release Notes

- Ability to use external PostgreSQL or Solr databases run locally or institutionally, rather than the included PostgreSQL and Solr databases.
- Many performance improvements in the repository, including paginated iteration and artifact caching.
- Upgraded the included Solr database from 6.6.5 to 7.2.1, and added tools to upgrade Solr across versions and schemas.
- REST services authenticate, and REST clients and the Pywb web replay engine provide credentials.
- Numerous bug fixes and performance improvements, and enhancements to the UI, installer, tests, documentation, and more.

Component Versions

LOCKSS 2.0.21-alpha2 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.21-alpha2
- LOCKSS Repository Service version 2.0.9.0
- LOCKSS Configuration Service version 2.0.3.0
- LOCKSS Poller Service version 2.0.1.0
- LOCKSS Metadata Extraction Service version 2.0.2.0
- LOCKSS Metadata Service version 2.0.1.0
- PostgreSQL version 9.6.12
- Apache Solr version 7.2.1
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

6.2 Network Ports

This page describes the default network ports used by the LOCKSS system.

Unless otherwise noted, all ports are **TCP**.

All ports in the 24600-24699 range should be considered reserved. The LCAP (LOCKSS polling and repair) port retains its historical value of 9729.

- 5432: PostgreSQL
- 8080: Pywb or OpenWayback replay engine
- 8983: Solr
- 9729: LCAP (LOCKSS polling and repair)
- 24600: *reserved* (currently LOCKSS Configuration Service UI)
- 24606: ActiveMQ
- 24610: LOCKSS Repository Service - REST port
- 24619: *reserved* (HDFS FS port)
- 24620: LOCKSS Configuration Service - Rest port

- 24621: LOCKSS Configuration Service - UI port
- 24630: LOCKSS Poller Service - REST port
- 24631: LOCKSS Poller Service - UI port
- 24640: LOCKSS Metadata Extraction Service - REST port
- 24641: LOCKSS Metadata Extraction Service - UI port
- 24650: LOCKSS Metadata Service - REST port
- 24651: LOCKSS Metadata Service - UI port
- 24670: LOCKSS Proxy
- 24671: *reserved*
- 24672: LOCKSS Audit Proxy
- 24673: *reserved*
- 24674: ICP server **(UDP)**
- 24680: LOCKSS Content Server (ServeContent)
- 24681: *reserved* Pywb replay engine
- 24682: *reserved* OpenWayback replay engine