

---

# **LOCKSS System Manual**

**LOCKSS Program**

**Sep 28, 2023**



# LOCKSS 2.0-ALPHA1 SYSTEM MANUAL

<b>1</b>	<b>Installing the LOCKSS System</b>	<b>3</b>
<b>2</b>	<b>Configuring the LOCKSS System</b>	<b>11</b>
<b>3</b>	<b>Running the LOCKSS System</b>	<b>13</b>
<b>4</b>	<b>Using the LOCKSS System</b>	<b>15</b>
<b>5</b>	<b>Appendix</b>	<b>21</b>



First release: 2.0.11-alpha1 (2019-05-13)

System manual last built: Sep 28, 2023

**Welcome to the LOCKSS 2.0-alpha1 System Manual.**



## 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.

## 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](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-1-3pu6hszjas19xyp7ghgosyx9k8atbfc8p2is99znp26u2lkl-
↪1awxwud3z9j1z3puu7rcgdbx \
        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 prefer, you can follow the [Local-Persist manual installation instructions](#).

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

- `sudo systemctl is-active docker-volume-local-persist` should say `active`
- `sudo systemctl is-enabled docker-volume-local-persist` should say `enabled`

Also verify that Local-Persist is registered with Docker:

- `sudo -u lockss docker info` should have a `Plugins` section with lists of volume, network and log plugins. The Volume list under the `Plugins` section should contain `local-persist`. Here is an excerpt of `docker info` output showing that Local-Persist is correctly registered as a Docker volume plugin:

### Plugins:

```
Volume: local local-persist
Network: bridge host macvlan null overlay
Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
```

## 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 Pystache

**Pystache** is the reference implementation of the **Mustache** templating language in Python. This page will walk you through the simple process of installing Pystache. The Pystache Python module depends on the Setuptools Python module.

You can install Pystache from *PyPI* <<https://pypi.org/>> using Pip:

```
sudo pip install setuptools pystache
```

Verify that Pystache is installed correctly:

```
pystache --help
```

You should see a help message about Pystache, for example:

```
Usage: pystache [-h] template context

Render a mustache template with the given context.

positional arguments:
  template      A filename or template string.
  context       A filename or JSON string.

Options:
  -h, --help  show this help message and exit
```

## 1.4 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.

### 1.4.1 Git on Arch Linux

Use Pacman to install Git on Arch Linux:

```
sudo pacman -S git
```

### 1.4.2 Git on CentOS

**Caution:** CentOS 7 is required.

Use Yum to install Git on CentOS:

```
sudo yum install git
```

### 1.4.3 Git on Debian

**Caution:** Debian 9 (Stretch) is required.

Use Apt to install Git on Debian:

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

### 1.4.4 Git on Fedora

**Caution:** Fedora 28 or better is required.

Use Dnf to install Git on Fedora:

```
sudo dnf install git
```

### 1.4.5 Git on Oracle Linux

**Caution:** Oracle Linux 7 is required.

Use Yum to install Git on Oracle Linux:

```
sudo yum install git
```

### 1.4.6 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
```

## 1.5 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
```

## 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`.)

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`. Questions include:

1. Fully qualified hostname (FQDN) of this machine: Enter the machine's hostname (e.g. `locksstest.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-alpha1; 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.
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).
19. **Password for database:** Enter the password for the embedded Postgres database included in LOCKSS 2.0-alpha. *Future versions will allow you to use an existing Postgres database and enter credentials accordingly.*
20. **Password for database (again):** Re-enter the password for the embedded Postgres database (if the two passwords do not match, the password will be asked again).
21. **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.

If prompted to generate files, accept (or run `scripts/generate-lockss` immediately after).



## 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:

- Run `scripts/generate-lockss`. This script takes your configuration data and turns it into a set of configuration files containing the right values.
- Run `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).
- Run `scripts/deploy-lockss`. This script deploys your LOCKSS stack by invoking Docker.



## USING THE LOCKSS SYSTEM

This section describes how to use the LOCKSS system.

### 4.1 Using the LOCKSS Configuration Service

---

**Note:** This page is under construction.

---

#### 4.1.1 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.

#### 4.1.2 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.

### 4.1.3 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.

### 4.1.4 Managing Access to the Web User Interfaces

*This section is under construction.*

## 4.2 Using the LOCKSS Metadata Extraction Service

---

**Note:** This page is under construction.

---

### 4.2.1 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.

### 4.2.2 Requesting Metadata Extraction

*This section is under construction.*

## 4.3 Using the LOCKSS Metadata Service

---

**Note:** This page is under construction.

---

### 4.3.1 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.

### 4.3.2 Requesting Metadata Information

*This section is under construction.*

## 4.4 Using the LOCKSS Poller Service

---

**Note:** This page is under construction.

---

### 4.4.1 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.

### 4.4.2 Requesting Polls

*This section is under construction.*

### 4.4.3 Monitoring Polling and Voting

*This section is under construction.*

## 4.5 Using the LOCKSS Crawler Service

---

**Note:** This page is under construction.

---

### 4.5.1 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.

### 4.5.2 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 successful 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.

### 4.5.3 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.

### 4.5.4 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.

## 4.6 Replaying Web Content with Pywb

### 4.6.1 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>:24681`.

### 4.6.2 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.

### 4.6.3 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.



## APPENDIX

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

### 5.1 Release Notes

#### 5.1.1 LOCKSS 2.0.11-alpha1

Released: 2019-05-13

Also known as: LOCKSS 2.0-alpha1a

LOCKSS 2.0.11-alpha1 (also known as LOCKSS 2.0-alpha1a) is the first release of the LOCKSS 2.0-alpha1 system.

#### Release Notes

The LOCKSS 2.0.11-alpha1 release features:

- Storage in the **LOCKSS Repository Service**, on top of [Solr](#).
- Node configuration and state management in the **LOCKSS Configuration Service**.
- Content ingest with the LOCKSS Crawler.
- LOCKSS polling and repair with the **LOCKSS Poller Service**.
- Metadata extraction and query in the **LOCKSS Metadata Extraction Service** and **LOCKSS Metadata Service**, on top of a [Postgres](#) database.
- Web replay with LOCKSS ServeContent and [Pywb](#), the state-of-the-art Web replay engine behind [Webrecorder](#).

#### Component Versions

LOCKSS 2.0.11-alpha1 consists of a configurable set of the following components:

- [LOCKSS Installer](#) version 2.0.11-alpha1
- [LOCKSS Repository Service](#) version 2.0.8.0
- [LOCKSS Configuration Service](#) version 2.0.1.0
- [LOCKSS Poller Service](#) version 2.0.0.0
- [LOCKSS Metadata Extraction Service](#) version 2.0.1.0

- LOCKSS Metadata Service version 2.0.0.0
- PostgreSQL version 9.6.12
- Apache Solr version 6.6.5
- Pywb version 2.2.20190410 (custom version 2.2.20190410-1)

## 5.2 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.

- 9729: LCAP (LOCKSS polling and repair)
- 24600: *reserved* (currently LOCKSS Configuration Service UI)
- 24602: PostgreSQL
- 24603: Solr
- 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: Pywb replay engine
- 24682: *reserved* OpenWayback replay engine