

---

# **LOCKSS System Manual**

**LOCKSS Program**

**2024-01-08**



# LOCKSS 2.0-ALPHA7 SYSTEM MANUAL

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	System Prerequisites . . . . .	1
1.2	Operating System . . . . .	2
<b>2</b>	<b>Upgrading the LOCKSS System</b>	<b>7</b>
2.1	Stop the LOCKSS System . . . . .	7
2.2	Update the LOCKSS Installer . . . . .	8
2.3	Run the Upgrade Script . . . . .	8
2.4	Re-run the Configure Script . . . . .	9
2.5	Start LOCKSS 2.0-alpha7 . . . . .	9
<b>3</b>	<b>Installing the LOCKSS System</b>	<b>11</b>
3.1	Creating the lockss User . . . . .	11
3.2	Downloading the LOCKSS Installer . . . . .	12
3.3	Running the LOCKSS Installer . . . . .	14
<b>4</b>	<b>Configuring the LOCKSS System</b>	<b>27</b>
4.1	Before Invoking <b>configure-lockss</b> . . . . .	27
4.2	Invoking <b>configure-lockss</b> . . . . .	28
4.3	Kubernetes Settings . . . . .	28
4.4	Network Settings . . . . .	28
4.5	Mail Settings . . . . .	29
4.6	Preservation Network Settings . . . . .	30
4.7	Storage Areas . . . . .	31
4.8	Web User Interface Settings . . . . .	32
4.9	Database Settings . . . . .	32
4.10	LOCKSS Services . . . . .	35
4.11	Web Replay Settings . . . . .	36
4.12	Final Steps . . . . .	36
<b>5</b>	<b>Running the LOCKSS System</b>	<b>37</b>
5.1	Starting the LOCKSS System . . . . .	37
5.2	Shutting down the LOCKSS System . . . . .	37
5.3	Restarting a Running LOCKSS System . . . . .	38
5.4	Removing a Configured LOCKSS System . . . . .	38
<b>6</b>	<b>Using the LOCKSS System</b>	<b>39</b>
6.1	Using the LOCKSS Configuration Service . . . . .	39
6.2	Using the LOCKSS Crawler Service . . . . .	40
6.3	Using the LOCKSS Poller Service . . . . .	42
6.4	Using the LOCKSS Metadata Extraction Service . . . . .	42

6.5	Using the LOCKSS Metadata Service . . . . .	43
6.6	Replaying Web Content with Pywb . . . . .	43
6.7	Replaying Web Content with OpenWayback . . . . .	44
6.8	Using the Kubernetes Dashboard . . . . .	45
<b>7</b>	<b>Troubleshooting the LOCKSS System</b>	<b>47</b>
7.1	Known Issues . . . . .	47
7.2	Troubleshooting <b>iptables</b> . . . . .	48
7.3	Troubleshooting <b>firewalld</b> . . . . .	49
7.4	Troubleshooting <b>ufw</b> . . . . .	50
7.5	Troubleshooting CoreDNS . . . . .	51
7.6	Troubleshooting the K3s Installer . . . . .	52
7.7	Troubleshooting the K3s Configuration Checker . . . . .	54
7.8	Troubleshooting OverlayFS with XFS . . . . .	56
<b>8</b>	<b>System Administration Tasks</b>	<b>59</b>
8.1	Running Commands as <b>root</b> . . . . .	59
8.2	Running Commands as a Privileged User . . . . .	59
8.3	Running Commands as the <b>lockss</b> User . . . . .	60
8.4	Operating System Updates . . . . .	61
8.5	Installing Curl . . . . .	64
8.6	Installing HTTPie . . . . .	69
8.7	Installing Wget . . . . .	74
8.8	Resetting the System to a Blank State . . . . .	79
<b>9</b>	<b>Appendix</b>	<b>81</b>
9.1	Security Advisories . . . . .	81
9.2	Release Notes . . . . .	81
9.3	Frequently Asked Questions . . . . .	85
9.4	Software License . . . . .	85
9.5	Network Ports . . . . .	86
9.6	Downloading the LOCKSS Installer using <b>git</b> . . . . .	87
9.7	Working with PostgreSQL . . . . .	87
9.8	LCAP Over SSL . . . . .	89

## INTRODUCTION

The LOCKSS system is a distributed digital preservation software system developed by the [LOCKSS Program](#), a division of the [Digital Library Systems and Services](#) department at [Stanford Libraries](#).

The 2.x series of the LOCKSS system stems from the LAAWS (LOCKSS Architected As Web Services) initiative, an ambitious modernization project that includes rewriting the classic LOCKSS daemon as a suite of containerized components, funded in part by a grant from the [Andrew W. Mellon Foundation](#).

This version, LOCKSS 2.0-alpha7, is the seventh "alpha" release of the LOCKSS system on the road to LOCKSS 2.0.

## 1.1 System Prerequisites

### 1.1.1 Host

The LOCKSS system runs in a **64-bit Linux** host (physical or virtual).

See the next section (*Operating System*) for operating system choices.

### 1.1.2 CPU

The CPU requirements depend on which components of the LOCKSS system you choose to run. We recommend at least **4 CPU cores**, preferably 8.

### 1.1.3 Memory

Likewise, the memory requirements also depend on which components of the LOCKSS system you choose to run. We recommend at least **16 GB** of memory for modest applications, more for machines involved in sizeable applications like the Global LOCKSS Network.

### 1.1.4 Storage

LOCKSS makes use of several storage areas. During configuration, the administrator must specify the location of these storage areas by supplying one or more directory paths. The default is to put all storage under a single directory, but different types of storage have different size and performance requirements and on a large system, if different types of storage are available it may be advantageous to place the storage areas on different devices:

- **Content storage:** This is where all the preserved content is stored.

The amount of space required depends on the amount of content that will be preserved. The content is efficiently stored in large, compressed WARC files. Unlike LOCKSS 1.x, inode usage is very low. Multiple content storage areas may be specified, and more can be added later.

- **State data storage:** This is used for databases (unless using external PostgreSQL and/or Solr) and other state data.
- **Temporary storage:** The LOCKSS software makes heavy use of temporary storage.

**Caution:** Depending on the characteristics of the preservation activities undertaken by the system, in some circumstances content processing may require a substantial amount of temporary space, up to tens of gigabytes. Do not use a RAM-based `tmpfs` volume, or a directory in a space-constrained partition, for temporary data storage.

- **Log storage:** Service logs will be written to subdirectories of this storage area.

### Network-Attached Storage

**We strongly discourage placing the state data and temporary storage areas on network-attached storage such as NFS, as performance will be severely impacted.** Likewise, we discourage placing the log storage area on network-attached storage such as NFS.

Additionally, many LOCKSS systems preserve a large amount of content and, while not recommended, some sites find it necessary to use network-attached storage such as NFS for the content storage area. LOCKSS' audit activities involve nearly continuous reading of preserved content from storage; performance of the LOCKSS system will be significantly impacted, and performance of the network storage subsystem may be impacted.

---

### What's the Minimum for Experimentation?

To review the installation instructions and test the installation of K3s in various operating systems, we routinely install and bring up minimal LOCKSS 2.0-alpha7, with no metadata services or Web replay engines, and with empty embedded Postgres and Solr databases, in Vagrant virtual machines with Virtualbox using 2 CPU cores and 3 GB of memory. These minimal VMs would not support a production load, but it can be a useful tool to try out the installation instructions or evaluate the system.

---

## 1.2 Operating System

The LOCKSS system requires a **64-bit Linux** host (physical or virtual) compatible with **K3s**, a lightweight Kubernetes distribution by **Rancher**. The K3s documentation states<sup>1</sup> that "K3s is expected to work on most modern Linux systems", and that "Some OSs have specific requirements" (which are documented in this manual and integrated into the **lockss-installer** scripts).

The LOCKSS team has successfully tested the LOCKSS system installation process on many flavors of Linux, some of which are listed below:

---

<sup>1</sup> Reference: <https://docs.k3s.io/installation/requirements#operating-systems>

## AlmaLinux OS

The LOCKSS system is compatible with [AlmaLinux OS](#):

Operating System	Version	Active Support	Security Support	Notes
AlmaLinux OS	9.1	2027-05-31	2032-05-31	
AlmaLinux OS	9.0	2027-05-31	2032-05-31	
AlmaLinux OS	8.7	2024-05-31	2029-05-31	
AlmaLinux OS	8.6	2024-05-31	2029-05-31	
AlmaLinux OS	8.5	2024-05-31	2029-05-31	
AlmaLinux OS	8.4	2024-05-31	2029-05-31	
AlmaLinux OS	8.3	2024-05-31	2029-05-31	

## Arch Linux

The LOCKSS system is compatible with [Arch Linux](#):

Operating System	Version	Active Support	Security Support	Notes
Arch Linux				rolling release

## CentOS

**Caution:** We no longer recommend CentOS for new installations; we recommend Rocky Linux instead.

The LOCKSS system is compatible with [CentOS](#):

Operating System	Version	Active Support	Security Support	Notes
CentOS Stream				rolling release
CentOS Linux	7.9	2020-08-06	2024-06-30	end of life
CentOS Linux	7.8	2020-08-06	2024-06-30	end of life
CentOS Linux	7.7	2020-08-06	2024-06-30	end of life
CentOS Linux	7.6	2020-08-06	2024-06-30	end of life
CentOS Linux	7.5	2020-08-06	2024-06-30	end of life
CentOS Linux	7.4	2020-08-06	2024-06-30	end of life
CentOS Linux	7.3	2020-08-06	2024-06-30	end of life

## Debian

The LOCKSS system is compatible with [Debian](#):

Operating System	Version	Active Support	Security Support	Notes
Debian	11.6	2024-07-01	2026-06-30	
Debian	11.5	2024-07-01	2026-06-30	
Debian	11.4	2024-07-01	2026-06-30	
Debian	11.3	2024-07-01	2026-06-30	
Debian	11.2	2024-07-01	2026-06-30	
Debian	11.1	2024-07-01	2026-06-30	
Debian	11.0	2024-07-01	2026-06-30	
Debian	10.13	2022-09-10	2024-06-30	end of life
Debian	10.12	2022-09-10	2024-06-30	end of life
Debian	10.11	2022-09-10	2024-06-30	end of life
Debian	10.10	2022-09-10	2024-06-30	end of life
Debian	10.9	2022-09-10	2024-06-30	end of life
Debian	10.8	2022-09-10	2024-06-30	end of life
Debian	10.7	2022-09-10	2024-06-30	end of life
Debian	10.6	2022-09-10	2024-06-30	end of life
Debian	10.5	2022-09-10	2024-06-30	end of life
Debian	10.4	2022-09-10	2024-06-30	end of life
Debian	10.3	2022-09-10	2024-06-30	end of life
Debian	10.2	2022-09-10	2024-06-30	end of life
Debian	10.1	2022-09-10	2024-06-30	end of life
Debian	10.0	2022-09-10	2024-06-30	end of life

## **EuroLinux**

The LOCKSS system is compatible with [EuroLinux](#):

Operating System	Version	Active Support	Security Support	Notes
EuroLinux	9.1	2032-05-31	2032-06-30	
EuroLinux	9.0	2032-05-31	2032-06-30	
EuroLinux	8.6	2029-03-01	2029-06-30	
EuroLinux	8.5	2029-03-01	2029-06-30	
EuroLinux	8.4	2029-03-01	2029-06-30	
EuroLinux	8.3	2029-03-01	2029-06-30	
EuroLinux	7.9	2024-07-31	2024-07-31	
EuroLinux	7.8	2024-07-31	2024-07-31	
EuroLinux	7.7	2024-07-31	2024-07-31	
EuroLinux	7.6	2024-07-31	2024-07-31	

## **Fedora Linux**

The LOCKSS system is compatible with [Fedora Linux](#):

Operating System	Version	Active Support	Security Support	Notes
Fedora Linux	37	2023-12-15	2023-12-15	
Fedora Linux	36	2023-05-16	2023-05-16	



## Linux Mint

The LOCKSS system is compatible with [Linux Mint](#):

Operating System	Version	Active Support	Security Support	Notes
Linux Mint	21	Yes	2027-04-01	
Linux Mint	20.3	Yes	2025-04-01	
Linux Mint	20.2	Yes	2025-04-01	
Linux Mint	20.1	No	2025-04-01	end of life
Linux Mint	20	No	2025-04-01	end of life

## OpenSUSE

The LOCKSS system is compatible with [OpenSUSE](#):

Operating System	Version	Active Support	Security Support	Notes
OpenSUSE Tumbleweed				rolling release
OpenSUSE Leap	15.4	2023-12-01	2023-12-01	

## Oracle Linux

The LOCKSS system is compatible with [Oracle Linux](#):

Operating System	Version	Active Support	Security Support	Notes
Oracle Linux	9.1	2032-07-01	2034-06-01	
Oracle Linux	9.0	2032-07-01	2034-06-01	
Oracle Linux	8.7	2029-07-01	2029-07-01	
Oracle Linux	8.6	2029-07-01	2029-07-01	
Oracle Linux	8.5	2029-07-01	2029-07-01	
Oracle Linux	8.4	2029-07-01	2029-07-01	
Oracle Linux	8.3	2029-07-01	2029-07-01	
Oracle Linux	8.2	2029-07-01	2029-07-01	
Oracle Linux	8.1	2029-07-01	2029-07-01	
Oracle Linux	7.9	2024-07-01	2026-06-01	
Oracle Linux	7.8	2024-07-01	2026-06-01	
Oracle Linux	7.7	2024-07-01	2026-06-01	
Oracle Linux	7.6	2024-07-01	2026-06-01	

## RHEL

The LOCKSS system is compatible with [RHEL](#):

Operating System	Version	Active Support	Security Support	Notes
RHEL	8.3	2024-05-31	2029-05-31	

### Rocky Linux

---

**Tip:** [Rocky Linux](#) is the operating system we currently recommend for new installations, and for existing installations based on CentOS or Scientific Linux.

---

The LOCKSS system is compatible with [Rocky Linux](#):

Operating System	Version	Active Support	Security Support	Notes
Rocky Linux	9.1	2025-05-31	2032-05-31	
Rocky Linux	9.0	2025-05-31	2032-05-31	
Rocky Linux	8.7	2024-05-31	2029-05-31	
Rocky Linux	8.6	2024-05-31	2029-05-31	
Rocky Linux	8.5	2024-05-31	2029-05-31	
Rocky Linux	8.4	2024-05-31	2029-05-31	

### Scientific Linux

**Caution:** We no longer recommend Scientific Linux for new installations; we recommend Rocky Linux instead.

The LOCKSS system is compatible with [Scientific Linux](#):

Operating System	Version	Active Support	Security Support	Notes
Scientific Linux	7.9	2024-06-30	2024-06-30	obsolescent
Scientific Linux	7.8	2024-06-30	2024-06-30	obsolescent
Scientific Linux	7.7	2024-06-30	2024-06-30	obsolescent
Scientific Linux	7.6	2024-06-30	2024-06-30	obsolescent

### Ubuntu

The LOCKSS system is compatible with [Ubuntu](#):

Operating System	Version	Active Support	Security Support	Notes
Ubuntu	22.10	2023-07-20	2023-07-20	
Ubuntu	22.04 LTS	2027-04-21	2032-04-01	
Ubuntu	20.04 LTS	2025-04-02	2030-04-01	
Ubuntu	18.04 LTS	2023-04-02	2028-04-01	

The LOCKSS system can likely be installed successfully on slightly different versions of the Linux flavors above, as well as other Linux flavors altogether, including commercial variants like [RHEL](#) or [SLES](#). We welcome reports of successful installations from the community so they can be added to the list above.

---

**Tip:** [Rocky Linux](#) is the operating system we currently recommend for new installations, and for existing installations based on CentOS or Scientific Linux.

---

## UPGRADING THE LOCKSS SYSTEM

---

**Note:** This chapter describes how to upgrade an existing LOCKSS 2.0-alpha5, 2.0-alpha6, or 2.0.71-alpha7 system to 2.0-alpha7. If you are installing the LOCKSS 2.x system for the first time, please see the installation instructions in the next chapter:

*Installing the LOCKSS System*

---

---

**Tip:** Before you begin the upgrade, we strongly recommend you first bring your operating system up to date by applying security updates and upgrading installed packages. Ask your system administrator or see *Operating System Updates* in the appendix.

---

---

**Note:** Commands in this section are run as the lockss user<sup>1</sup>.

---

### 2.1 Stop the LOCKSS System

The first step is to stop the running LOCKSS system. Log in as the lockss user and run the following command in the *LOCKSS Installer Directory*:

```
scripts/stop-lockss
```

You may verify all LOCKSS components have stopped by running the following command:

```
k3s kubectl get deployments -n lockss
```

which should return:

```
No resources found in lockss namespace.
```

---

<sup>1</sup> See *Running Commands as the lockss User*.

## 2.2 Update the LOCKSS Installer

As the lockss user<sup>Page 7, 1</sup>, run this Curl, Wget or HTTPie command<sup>2</sup>:

### Curl

```
curl -sSfL https://lockss.org/downloader | sh -s -
```

### HTTPie

```
http -qd https://lockss.org/downloader | sh -s -
```

### Wget

```
wget -q0- https://lockss.org/downloader | sh -s -
```

This will download and invoke the LOCKSS Downloader, which in turn will install the latest version of the LOCKSS Installer into the *Default LOCKSS Installer Directory* (`$HOME/lockss-installer`). If you are using a custom LOCKSS Installer Directory *DIR*, remember to add `--download-dir=DIR` to the end of the command; see *Running the LOCKSS Downloader* for details.

## 2.3 Run the Upgrade Script

The next step will update PostgreSQL from 9.6.12 to 14.7 if applicable. In addition, if updating from 2.0-alpha5, the archived content will then be reindexed. As the lockss user, run the following command in the *LOCKSS Installer Directory*:

```
scripts/upgrades/upgrade-to-alpha7
```

---

### Hint:

If it takes more than a few seconds for `upgrade-to-alpha7` above to run, the reindexing of all previously archived content which occurs the first time you start 2.0-alpha7 after upgrading from 2.0-alpha5 may take prohibitively long. This performance issue will be addressed in a later release. If you do not need the previously stored content during alpha testing, you could delete it and skip this reindexing step; see *Resetting the System to a Blank State*.

---

<sup>2</sup> Most typical Linux systems have at least one of Curl, Wget or HTTPie installed by default. You can check by typing `curl --version`, `wget --version` or `http --version`, and seeing which ones do not output an error message. See *Installing Curl*, *Installing Wget* or *Installing HTTPie* for installation instructions.

## 2.4 Re-run the Configure Script

Re-run the configuration script by running the command below and follow the instructions in [Configuring the LOCKSS System](#) to ensure all existing configuration parameters are still correct and to configure any new parameters:

```
scripts/configure-lockss -r
```

The `-r` ("replay") option will re-use all previously-entered configuration values, and only ask questions for new prompts added since the previous release.

## 2.5 Start LOCKSS 2.0-alpha7

You are now ready to start the LOCKSS system.

- If you were upgrading from LOCKSS 2.0.71-alpha7, run this command:

```
scripts/start-lockss -u
```

- If you were upgrading from LOCKSS 2.0-alpha5 or LOCKSS 2.0-alpha6, run this command:

```
scripts/start-lockss
```

---

**Hint:** If it takes more than a few seconds for `upgrade-to-alpha7` above to run, the reindexing of all previously archived content which occurs the first time you start 2.0-alpha7 after upgrading from 2.0-alpha5 may take prohibitively long. This performance issue will be addressed in a later release. If you do not need the previously stored content during alpha testing, you could delete it and skip this reindexing step; see [Resetting the System to a Blank State](#).

---



## INSTALLING THE LOCKSS SYSTEM

---

**Note:** This chapter describes how to install the LOCKSS 2.0-alpha7 system from scratch. If you are upgrading an existing LOCKSS 2.0-alpha5, 2.0-alpha6 or 2.0.71-alpha7 system to 2.0.72-alpha7, please see the upgrade instructions in the previous chapter:

*Upgrading the LOCKSS System*

---

---

**Tip:** Before you begin installing the LOCKSS system, we strongly recommend you first bring your operating system up to date by applying security updates and upgrading installed packages. Ask your system administrator or see *Operating System Updates*.

---

### 3.1 Creating the lockss User

---

**Note:** Commands in this section are run as `root`<sup>1</sup>.

---

The first task is to create a system user named `lockss`, under which the LOCKSS system will run.

Run this **useradd** command as `root`<sup>Page 11, 1</sup>:

```
useradd --system --user-group --create-home --shell=/bin/bash lockss
```

or equivalently:

```
useradd -rUms /bin/bash lockss
```

This will create a `lockss` system user, a `lockss` system group, and a home directory in `/home/lockss`.

---

---

<sup>1</sup> See *Running Commands as root*.

## 3.2 Downloading the LOCKSS Installer

---

**Note:** Commands in this section are run as the `lockss` user<sup>1</sup>.

---

The next task is to download the LOCKSS Installer.

### 3.2.1 LOCKSS Installer Directory

The directory into which the LOCKSS Installer is downloaded is simply known as the **LOCKSS Installer Directory**.

Many commands in this manual, such as those to install, configure, start and stop the LOCKSS system, are relative to the LOCKSS Installer Directory, meaning you need to navigate to it at the console before issuing the relevant command.

### 3.2.2 Default LOCKSS Installer Directory

The **default LOCKSS Installer Directory** will be `${HOME}/lockss-installer` (where `${HOME}` refers to the `lockss` user's home directory, typically `/home/lockss`), unless you use the LOCKSS Downloader's `--download-dir` option to set a custom LOCKSS Installer Directory.

### 3.2.3 Running the LOCKSS Downloader

To download the LOCKSS Installer, you will use [Curl](#), [Wget](#) or [HTTPIe](#)<sup>2</sup> to invoke the LOCKSS Downloader<sup>3</sup>, whose default action is to download the LOCKSS Installer.

As the `lockss` user<sup>Page 12, 1</sup>, run this Curl, Wget or HTTPIe command<sup>2</sup>:

#### Curl

```
curl -sSfL https://lockss.org/downloader | sh -s -
```

#### HTTPIe

```
http -qd https://lockss.org/downloader | sh -s -
```

---

<sup>1</sup> See *Running Commands as the lockss User*.

<sup>2</sup> Most typical Linux systems have at least one of [Curl](#), [Wget](#) or [HTTPIe](#) installed by default. You can check by typing `curl --version`, `wget --version` or `http --version`, and seeing which ones do not output an error message. See *Installing Curl*, *Installing Wget* or *Installing HTTPIe* for installation instructions.

<sup>3</sup> The LOCKSS Downloader is a script to download GitHub projects without Git, with Curl, Wget or HTTPIe instead. See <https://github.com/lockss/lockss-downloader>.



## Wget

```
wget -q0- https://lockss.org/downloader | sh -s -
```

This will download the LOCKSS Installer into the *Default LOCKSS Installer Directory*.

**Tip:** Below are some advanced tips for this section.

## Inspecting the LOCKSS Downloader before running it

For security purposes, you may wish to inspect the LOCKSS Downloader before executing it.

One option is to review the contents of the script directly on GitHub to your satisfaction, then execute it as described above. The URL <https://lockss.org/downloader> redirects to <https://github.com/lockss/lockss-downloader/raw/main/lockss-downloader>.

Another option is to download a copy of the LOCKSS Downloader script, review it, then execute it, all locally. To do so, follow this procedure:

1. As the lockss user<sup>1</sup>, run this Curl, Wget or HTTPie command<sup>Page 12, 2</sup>:

### Curl

```
curl -Lo lockss-downloader https://lockss.org/downloader
```

### HTTPie

```
http -qdo lockss-downloader https://lockss.org/downloader
```

### Wget

```
wget -q0 lockss-downloader https://lockss.org/downloader
```

This will download the LOCKSS Downloader script into the current directory as `lockss-downloader`.

2. Inspect the file `lockss-downloader` to your satisfaction.
3. Run this command:

```
chmod +x lockss-downloader
```

to make the LOCKSS Downloader script executable.

4. Type:

```
./lockss-downloader
```

to run the LOCKSS Downloader script. You can append to `./lockss-downloader` all the same options that can be appended to `| sh -s -` in the normal procedure documented in this section, for instance `./lockss-downloader --download-dir=DIR`.

### Custom LOCKSS Installer Directory

If you need your *LOCKSS Installer Directory* to be a directory *DIR* other than the *Default LOCKSS Installer Directory*, add `--download-dir=DIR` (or `-d DIR`) after `| sh -s -`, like so:

```
... | sh -s - --download-dir=DIR
```

### Custom version of the LOCKSS Installer

If you have a reason to install a version of the LOCKSS Installer other than the latest stable release 2.0.72-alpha7, you can do so by making references to the `lockss-installer` Git repository on GitHub<sup>4</sup>:

- You can install a version from the tip of a given branch *BRA* of the `lockss-installer` Git repository (e.g. `develop`) by adding `--git-branch=BRA` (or `-b BRA`) after `| sh -s -`. This might be needed if you are helping the LOCKSS Team test a development, pre-release, or hotfix version of the LOCKSS Installer.
- You can install a version labeled by a given tag *TAG* of the `lockss-installer` Git repository (e.g. `version-2.0.61-alpha6`) by adding `--git-tag=TAG` (or `-t TAG`) after `| sh -s -`. This might be needed if you are installing a specific past version of the LOCKSS Installer.
- You can install a version as of a specific commit identifier *COM* of the `lockss-installer` Git repository by adding `--git-commit=COM` (or `-c COM`) after `| sh -s -`. This might be needed if you are helping the LOCKSS Team test a development version of the LOCKSS Installer.

### Considerations if using `sudo -u`

If you must use:

```
... | sudo -u lockss sh -s -
```

to invoke the LOCKSS Downloader as the `lockss` user, beware that *typically* it will run in a context where `${HOME}` has been adjusted to the home directory of the `lockss` user, but this is *not guaranteed* -- it depends on the way **sudo** is configured on your host system. To *ensure* `${HOME}` is set correctly, use the `-H` (`--set-home`) option of **sudo**, for example like so:

```
... | sudo -Hu lockss sh -s -
```

---

## 3.3 Running the LOCKSS Installer

---

**Note:** Commands in this section are run as `root`<sup>1</sup>.

---

The next task is to run the LOCKSS Installer.

The installation process goes through various phases:

- Checking that some prerequisites to install K3s are met. No user interaction is expected.

---

<sup>4</sup> See <https://github.com/lockss/lockss-installer>.

<sup>1</sup> See *Running Commands as root*.

- Checking that the lockss system user and group exist. No user interaction is expected.
- Configuring **iptables**, **firewalld** and **ufw** for K3s. If applicable, you will be prompted to confirm before your system configuration is modified. You may incidentally be prompted for your **sudo** password.
- Configuring CoreDNS for K3s. If applicable, you will be prompted to enter non-loopback IP addresses of DNS servers.
- Installing K3s. If applicable, you will be prompted for a Kubernetes state data storage directory.
- Testing the K3s node. No user interaction is expected.

After the LOCKSS Installer succeeds, you can also optionally run the K3s Configuration Checker.

### 3.3.1 Invoking the LOCKSS Installer

To start the installation process, run this command (relative to the *LOCKSS Installer Directory*) as **root**<sup>Page 14, 1</sup>:

```
scripts/install-lockss
```

The installer will run through its phases, each of which is described in its own section below from *Checking K3s Prerequisites* (Section 3.3.2) to *Completion of the LOCKSS Installation Process* (Section 3.3.10).

**Tip:** Below are some advanced tips for this section.

#### Skipping install-lockss phases

You may need to skip some of the phases of **install-lockss**, for example to overcome an incompatibility with the specifics of your host system. If this is necessary, invoke **install-lockss** with one or more of the following options:

Option	Phase(s) skipped
<b>--skip-check-prerequisites</b>	<i>Checking K3s Prerequisites</i> (Section 3.3.2)
<b>--skip-check-system-user</b>	<i>Checking the System User and Group</i> (Section 3.3.3)
<b>--skip-configure-iptables</b>	<i>Configuring iptables for K3s</i> (Section 3.3.4)
<b>--skip-configure-firewalld</b>	<i>Configuring firewalld for K3s</i> (Section 3.3.5)
<b>--skip-configure-ufw</b>	<i>Configuring ufw for K3s</i> (Section 3.3.6)
<b>--skip-configure-coredns</b>	<i>Configuring CoreDNS for K3s</i> (Section 3.3.7)
<b>--skip-install-k3s</b>	<ul style="list-style-type: none"> <li>• <i>Checking K3s Prerequisites</i> (Section 3.3.2)</li> <li>• <i>Configuring iptables for K3s</i> (Section 3.3.4)</li> <li>• <i>Configuring firewalld for K3s</i> (Section 3.3.5)</li> <li>• <i>Configuring ufw for K3s</i> (Section 3.3.6)</li> <li>• <i>Configuring CoreDNS for K3s</i> (Section 3.3.7)</li> <li>• <i>Installing K3s</i> (Section 3.3.8)</li> </ul>
<b>--skip-test-k3s</b>	<i>Testing the K3s Node</i> (Section 3.3.9)

When a phase is skipped as a result of one of these options, you will see a message similar to this during the corresponding phase:

```
[success] Skipping (--skip-configure-firewalld)
```

## Running only one install-lockss phase

Conversely, you may need to run or re-run only one phase of **install-lockss**, for example re-running the *Testing the K3s Node* phase after it fails and you perform some troubleshooting. If this is necessary, invoke **install-lockss** with exactly one of the following options:

Option	Phase executed
<code>--check-prerequisites</code> (or <code>-P</code> )	<i>Checking K3s Prerequisites</i> (Section 3.3.2)
<code>--check-system-user</code> (or <code>-L</code> )	<i>Checking the System User and Group</i> (Section 3.3.3)
<code>--configure-iptables</code> (or <code>-I</code> )	<i>Configuring iptables for K3s</i> (Section 3.3.4)
<code>--configure-firewalld</code> (or <code>-F</code> )	<i>Configuring firewalld for K3s</i> (Section 3.3.5)
<code>--configure-ufw</code> (or <code>-U</code> )	<i>Configuring ufw for K3s</i> (Section 3.3.6)
<code>--configure-coredns</code> (or <code>-C</code> )	<i>Configuring CoreDNS for K3s</i> (Section 3.3.7)
<code>--install-k3s</code> (or <code>-K</code> )	<i>Installing K3s</i> (Section 3.3.8)
<code>--test-k3s</code> (or <code>-T</code> )	<i>Testing the K3s Node</i> (Section 3.3.9)

## Running install-lockss on auto-pilot

If you invoke **install-lockss** with the `--assume-yes` (or `-y`) option, it will attempt to run without asking any questions interactively, by assuming that the answer to any yes/no question is "yes" and that the answer to other interactive questions is the suggested default value. **This is only appropriate for advanced users** who understand the implications of the default code paths in *Configuring iptables for K3s* (Section 3.3.4), *Configuring firewalld for K3s* (Section 3.3.5), *Configuring ufw for K3s* (Section 3.3.6), *Configuring CoreDNS for K3s* (Section 3.3.7) and *Installing K3s* (Section 3.3.8) on the host system, for example after previous experience installing the LOCKSS system.

---

## 3.3.2 Checking K3s Prerequisites

During this phase, **install-lockss** will check that certain prerequisites to installing K3s are met. This phase begins with this heading:

Checking K3s prerequisites...

No user interaction is expected; if everything goes well, you will see this message:

[success] K3s prerequisites checked

and **install-lockss** will successfully proceed to the next phase, *Checking the System User and Group* (Section 3.3.3).

**Error:** Below are some error conditions you may encounter here and what to do about them.

### User namespaces must be enabled in RHEL/CentOS 7

In some RHEL 7 and CentOS 7 systems, user namespaces are not enabled by default. If this is the case, you will see the error message:

[ERROR] User namespaces must be enabled in RHEL/CentOS 7; see manual

and **install-lockss** will fail. See *Enabling User Namespaces in RHEL 7 and CentOS 7* for troubleshooting, then go back to *Invoking the LOCKSS Installer* to try again.

**Apparmor enabled but apparmor\_parser missing**

In some systems, Apparmor is enabled but **apparmor\_parser** is not installed. If this is the case, you will see the error message:

```
[ERROR] apparmor enabled but apparmor_parser missing; see manual
```

and **install-lockss** will fail. See *Installing apparmor\_parser* for troubleshooting, then go back to *Invoking the LOCKSS Installer* to try again.

### 3.3.3 Checking the System User and Group

During this phase, **install-lockss** will check that the lockss user and group exist on the host system. This phase begins with the heading:

```
Checking the system user and group...
```

No user interaction is expected; if everything goes well, you will see this message:

```
[success] System user and group present
```

and **install-lockss** will successfully proceed to the next phase, *Configuring iptables for K3s* (Section 3.3.4).

**Error:** Below are some error conditions you may encounter here and what to do about them.

**lockss user or group does not exist**

If the lockss user or group does not exist on the host system, you will see one of these error messages:

```
[ERROR] The lockss user does not exist
```

```
[ERROR] The lockss group does not exist
```

and **install-lockss** will fail. Go back to the *Creating the lockss User* section to create the lockss user and group, then return to *Invoking the LOCKSS Installer* to try again.

### 3.3.4 Configuring iptables for K3s

During this phase, **install-lockss** will configure **iptables** to work with K3s, if applicable. This phase begins with the heading:

```
Configuring iptables for K3s...
```

In many situations, no configuration of **iptables** is needed; you will see one of these messages:

```
[success] Skipping (iptables is not on the PATH nor run via Alternatives)
```

```
[success] Skipping (iptables version is older than 1.8.0)
```

```
[success] Skipping (iptables version is newer than 1.8.3)
```

(continues on next page)

(continued from previous page)

```
[success] Skipping (iptables is in legacy mode)
[success] Skipping (iptables is not run via Alternatives)
```

and **install-lockss** will successfully proceed to the next phase, *Configuring firewalld for K3s* (Section 3.3.5).

Otherwise, you will receive the following prompt:

*Switch iptables to legacy mode via Alternatives?*

Enter Y to accept the proposed **iptables** configuration, or enter N to bypass, or hit **Enter** to accept the default in square brackets<sup>2</sup>. (You may be prompted for your **sudo** password.)

**Caution:** If you choose to bypass the proposed **iptables** configuration, you will see the warning:

```
[Warning] Leaving iptables unchanged; see manual for details
```

and **install-lockss** will keep going. But K3s may malfunction without further intervention; see *Troubleshooting iptables* for details.

**Error:** Below are some error conditions you may encounter here and what to do about them.

#### **iptables configuration attempt fails**

If the **iptables** configuration attempt fails, you will see one of these error messages:

```
[ERROR] Error deactivating ufw
[ERROR] Error applying update-alternatives to iptables
[ERROR] Error applying update-alternatives to ip6tables
[ERROR] Error flushing iptables
[ERROR] Error reactivating ufw
```

and **install-lockss** will fail. See *Troubleshooting iptables* for remediation details.

### 3.3.5 Configuring firewalld for K3s

During this phase, **install-lockss** will configure **firewalld** to work with K3s, if applicable. This phase begins with the heading:

```
Configuring firewalld for K3s...
```

In many situations, no configuration of **firewalld** is needed; you will see one of these messages:

<sup>2</sup> If **install-lockss** was invoked with the `--assume-yes` option, Y is automatically entered for you.

```
[success] Skipping (firewall-cmd is not on the PATH)
```

```
[success] Skipping (firewalld is not running)
```

and **install-lockss** will successfully proceed to the next phase, *Configuring ufw for K3s* (Section 3.3.6).

Otherwise, you will receive the following prompt:

*Add 10.42.0.0/16 and 10.43.0.0/16 to firewalld's trusted zone?*

Enter Y to accept the proposed **firewalld** configuration, or enter N to bypass, or hit **Enter** to accept the default in square brackets<sup>Page 18, 2</sup>. (You may be prompted for your **sudo** password.)

**Caution:** If you choose to bypass the proposed **firewalld** configuration, you will see the warning:

```
[Warning] Leaving firewalld unchanged; see manual for details
```

and **install-lockss** will keep going. But K3s may malfunction without further intervention; see *Troubleshooting firewalld* for details.

**Error:** Below are some error conditions you may encounter here and what to do about them.

#### **firewalld configuration attempt fails**

If the **firewalld** configuration attempt fails, you will see one of these error messages:

```
[ERROR] Could not add 10.42.0.0/16 to firewalld's trusted zone
```

```
[ERROR] Could not add 10.43.0.0/16 to firewalld's trusted zone
```

```
[ERROR] Could not reload firewalld
```

and **install-lockss** will fail. See *Troubleshooting firewalld* for remediation details.

### 3.3.6 Configuring ufw for K3s

During this phase, **install-lockss** will configure **ufw** to work with K3s, if necessary. This phase begins with the heading:

```
Configuring ufw for K3s...
```

In many situations, no configuration of **firewalld** is needed; you will see one of these messages:

```
[success] Skipping (ufw is not on the PATH)
```

```
[success] Skipping (ufw is not active)
```

and **install-lockss** will successfully proceed to the next phase, *Configuring CoreDNS for K3s* (Section 3.3.7).

Otherwise, you will receive the following prompt:

*Allow traffic from 10.42.0.0/16 and 10.43.0.0/16 via ufw?*

Enter Y to accept the proposed **ufw** configuration, or enter N to bypass, or hit **Enter** to accept the default in square brackets<sup>Page 18, 2</sup>. (You may be prompted for your **sudo** password.)

**Caution:** If you choose to bypass the proposed **ufw** configuration, you will see the warning:

```
[Warning] Leaving ufw unchanged; see manual for details
```

and **install-lockss** will keep going. But K3s may malfunction without further intervention. See *Troubleshooting ufw* for details.

**Error:** Below are some error conditions you may encounter here and what to do about them.

### **ufw** configuration attempt fails

If the **ufw** configuration attempt fails, you will see one of these error messages:

```
[ERROR] Could not allow traffic from 10.42.0.0/16 via ufw
```

```
[ERROR] Could not allow traffic from 10.43.0.0/16 via ufw
```

```
[ERROR] Could not reload ufw
```

and **install-lockss** will fail. See *Troubleshooting ufw* for remediation details.

## 3.3.7 Configuring CoreDNS for K3s

During this phase, **install-lockss** will configure CoreDNS to work with K3s, if necessary. This phase begins with the heading:

```
Configuring CoreDNS for K3s...
```

In many situations, no configuration of **firewalld** is needed; you will see this message:

```
[success] Using system resolv.conf files
```

and **install-lockss** will successfully proceed to the next phase, *Installing K3s* (Section 3.3.8).

Otherwise<sup>4</sup>, you will receive a message including CoreDNS does not allow a loopback address to be given to Kubernetes pods as an upstream DNS server, and the following prompt:

*IP address(es) of DNS resolvers, separated by ';'.*

Enter a semicolon-separated list of DNS server IP addresses that are *not* loopback addresses. A suggested value will be offered to you in square brackets, consisting of non-loopback IP addresses collected from your machine's DNS configuration; you can simply hit **Enter** to accept the suggested value<sup>3</sup>.

**Error:** Below are some error conditions you may encounter here and what to do about them.

---

<sup>4</sup> Or if your **install-lockss** was invoked with the `--force-dns-prompt` option.

<sup>3</sup> If **install-lockss** was invoked with the `--assume-yes` option, the suggested value is automatically accepted for you.



**CoreDNS configuration attempt fails**

If the CoreDNS configuration attempt fails, you will see one of these error messages:

```
[ERROR] Could not create /etc/lockss
```

```
[ERROR] Error rendering config/templates/k3s/resolv.conf.mustache to config/resolv.  
→ conf
```

```
[ERROR] Could not copy config/resolv.conf to /etc/lockss/resolv.conf
```

and **install-lockss** will fail. See [Troubleshooting CoreDNS](#) for remediation details.

### 3.3.8 Installing K3s

During this phase, **install-lockss** will install K3s 1.21.5+k3s1, if applicable. This phase begins with the heading:

```
Installing K3s...
```

This phase consists of these steps:

1. First, **install-lockss** will determine if K3s 1.21.5+k3s1 needs to be installed:

- If K3s is not present, **install-lockss** will display *K3s is not present*, and *will* install K3s 1.21.5+k3s1 in the next step.
- If an older version of K3s is present, **install-lockss** will display *Detected K3s version <installed\_version> is older than expected version <expected\_version>*, and you will receive the following prompt:

```
Upgrade K3s from <installed_version> to <expected_version>?
```

Enter Y and **install-lockss** will install K3s 1.21.5+k3s1 in the next step, or enter N and **install-lockss** will not install K3s 1.21.5+k3s1 in the next step, or hit Enter to accept the default in square brackets<sup>Page 18, 2</sup>.

- If the expected version of K3s is already present, **install-lockss** will display *K3s version <installed\_version> is already installed; skipping*, and *will not* install K3s 1.21.5+k3s1 in the next step.
  - If a more recent version of K3s is present, **install-lockss** will display *Detected K3s version <installed\_version> is more recent than expected version <expected\_version>*, and *will not* install K3s 1.21.5+k3s1 in the next step.
  - If K3s is detected but the installed and expected version numbers cannot be compared automatically, **install-lockss** will display *[Warning] Detected K3s version <installed\_version>, expected version <expected\_version>, comparison failure, skipping*, and **install-lockss** will not install K3s in the next step.
2. If **install-lockss** determined in the previous step that it *will not* install K3s 1.21.5+k3s1, you will see the confirmation *Not installing K3s*, and nothing will happen in this step.
- But if **install-lockss** determined in the previous step that it *will* install K3s 1.21.5+k3s1, you will see the confirmation *Installing K3s version <expected\_version>*, and this step will proceed as follows:
- a. First, **install-lockss** will ask you to specify the K3s state data directory (the directory K3s uses to store state data), with this prompt:
- ```
K3s state data directory
```

By default, this is `/var/lib/rancher/k3s`. However, if `/var` is space-limited, you should specify a different directory, that has ample space and is not backed by NFS or by XFS with legacy `ftype=0`.

Enter a suitable directory path for the K3s state data directory, or hit **Enter** to accept the default in square brackets<sup>Page 20, 35</sup>.

- b. Then **install-lockss** will attempt to determine the filesystem type of the specified K3s state data directory. In many situations, it will simply display the filesystem type in a message similar to this (for example, `<fs_type>` might be `ext4`):

Filesystem type of `<k3s_dir>` (`<k3s_mountpoint>`) is `<fs_type>`; proceeding

**Error:** Below are some error conditions you may encounter here and what to do about them.

#### Filesystem type of K3s state data directory is NFS

If the filesystem type backing the K3s state data directory is NFS, you will see the error message:

[ERROR] Filesystem type of `<k3s_dir>` (`<k3s_mountpoint>`) is NFS; see manual and **install-lockss** will fail. It is not possible to run K3s with a state data directory backed by NFS<sup>6</sup>. Re-run **install-lockss** and designate a different K3s state data directory that is not backed by NFS.

#### Filesystem type of K3s state data directory is XFS with legacy `ftype=0`

If the filesystem type backing the K3s state data directory is XFS with legacy `ftype=0`, you will see the error message:

[ERROR] Filesystem type of `<k3s_dir>` (`<k3s_mountpoint>`) is XFS with legacy `ftype=0`; see manual for workaround

and **install-lockss** will fail. Contemporary XFS filesystems with modern `ftype=1` work well with K3s, but older XFS filesystems with legacy `ftype=0` are not compatible. Ideally, re-run **install-lockss** and designate a different K3s state data directory that is not backed by XFS with legacy `ftype=0`. Alternatively, you can read about a workaround in [Troubleshooting OverlayFS with XFS](#).

**Warning:** Below are some warning messages you may see here and how to respond to them.

#### Filesystem type of K3s state data directory unknown

If the filesystem type backing the K3s state data directory cannot be inferred automatically, you will see the warning:

[Warning] Filesystem type of `<k3s_dir>` unknown (findmnt not present); proceeding

and **install-lockss** will keep going. But K3s may malfunction if the actual filesystem type backing the selected K3s state data directory is one that does not work with K3s, such as NFS, or XFS with legacy `ftype=0`; see the error conditions above.

<sup>5</sup> If **install-lockss** was invoked with the `--k3s-data-dir=DIR` option, `DIR` will automatically be used without the prompt.

<sup>6</sup> See <https://github.com/containerd/containerd/discussions/6140>.

**Filesystem type of K3s state data directory is XFS but ftype unknown**

If the `ftype` of the XFS filesystem backing the K3s state data directory cannot be inferred automatically, you will see the warning:

```
[Warning] Filesystem type of k3s_dir (k3s_mountpoint) is XFS but ftype
unknown (xfs_info not present); proceeding
```

and **install-lockss** will keep going. But K3s may malfunction if the actual filesystem type backing the selected K3s state data directory is XFS with legacy `ftype=0`; see the corresponding error condition above.

- c. Then **install-lockss** will download the K3s Installer from <https://get.k3s.io/> and invoke it with suitable options. This may take several minutes, during which the output to the console will be from the K3s Installer, not from **install-lockss**.

Depending on your operating system and other factors, the K3s Installer may install additional software packages or configure system components, using **sudo** if necessary (which may prompt for the user's **sudo** password).

**Error:** If the K3s Installer does not succeed, it will display its own error messages, then **install-lockss** will fail. See [Troubleshooting the K3s Installer](#) for remediation details.

**Sample error messages from the K3s Installer**

Error messages that the K3s Installer may display include:

```
[ERROR] Failed to apply container_runtime_exec_t to /usr/local/bin/k3s,
↳ please install:
    yum install -y container-selinux selinux-policy-base
    yum install -y https://rpm.rancher.io/k3s/stable/common/centos/8/noarch/
↳ k3s-selinux-0.3-0.el8.noarch.rpm
```

```
Error: Package: k3s-selinux-0.3-0.el7.noarch (rancher-k3s-common-stable)
Requires: container-selinux >= 2.107-3
You could try using --skip-broken to work around the problem
You could try running: rpm -Va --nofiles --nodigest
```

3. Finally, whether or not K3s was installed, **install-lockss** will store Kubernetes configuration data as the lockss user in the file `config/k8s.cfg` (relative to the [LOCKSS Installer Directory](#)).

**Error:** Below are some error conditions you may encounter here and what to do about them.

**Could not write or append to k8s.cfg**

If the creation of the file fails, you will see one of these error messages:

```
[ERROR] Could not write k8s.cfg
```

```
[ERROR] Could not append to k8s.cfg
```

and **install-lockss** will fail. Check for file permission mismatches between the user running **install-lockss** and the lockss-installer/config directory, then try again.

### 3.3.9 Testing the K3s Node

During this phase, **install-lockss** runs a series of tests to verify that the K3s node is operational. This phase begins with the heading:

```
Testing the K3s node...
```

No user interaction is expected. If all tests pass, you will see the message:

```
[success] Tested the K3s node
```

and **install-lockss** will successfully proceed to the next phase, *Completion of the LOCKSS Installation Process* (Section 3.3.10).

Otherwise, you will see an error message corresponding to the test that did not pass, and **install-lockss** will fail.

**Error:** Below are some error conditions you may encounter here and what to do about them.

#### Problems with config/k8s.cfg

At the end of Section 3.3.8 (*Installing K3s*), some Kubernetes-related data is stored in config/k8s.cfg (relative to the *LOCKSS Installer Directory*). If the file cannot be found or read, or if it contains invalid or unexpected data, you may see one of these error messages:

```
[ERROR] k8s.cfg not found

[ERROR] Error reading K8S_FLAVOR

[ERROR] K8S_FLAVOR is not set

[ERROR] K8S_FLAVOR is not k3s

[ERROR] Error reading KUBECTL_CMD

[ERROR] KUBECTL_CMD is not set

[ERROR] k3s command of KUBECTL_CMD is not on the PATH
```

Check the contents of config/k8s.cfg and contact us () for troubleshooting if necessary.

#### Problems with the K3s node

If the K3s node is not behaving as expected, you may see one of these errors:

```
[ERROR] Command failed (kubectl version)

[ERROR] Timeout waiting for the K3s node to be ready

[ERROR] Command failed (kubectl get node)

[ERROR] Unexpected number of K3s nodes
```

If the K3s node is newly installed, it may simply be that there has not yet been enough time for it to come up; you can re-run this phase with `scripts/install-lockss --test-k3s` (or `scripts/install-lockss -K`) to retry. Contact us () for troubleshooting if necessary.

### Problems with DNS

If the K3s node's DNS environment is not working properly, you may see one of these errors:

```
[ERROR] Timeout waiting for the CoreDNS pod to be running and ready

[ERROR] Command failed (kubectl get pod)

[ERROR] Unexpected number of CoreDNS pods

[ERROR] Timeout waiting for the DNS service to be present

[ERROR] Command failed (kubectl get service)

[ERROR] Unexpected number of kube-dns services

[ERROR] Unexpected kube-dns service type

[ERROR] Timeout waiting for DNS resolution

[ERROR] Unexpected Cluster-IP
```

If the K3s node is newly installed, it may simply be that there has not yet been enough time for CoreDNS to come up; you can re-run this phase with `scripts/install-lockss --test-k3s` (or `scripts/install-lockss -T`) to retry. You can also use the **install-lockss** options `--retries=N` (to increase the number of retries in DNS lookup tests to *N* from 5) or `--wait=S` (to increase the delay between retries in DNS lookup tests to *S* seconds from 10). Contact us () for troubleshooting if necessary.

### 3.3.10 Completion of the LOCKSS Installation Process

If all phases completed successfully, you will see the message:

```
[success] Successful completion of the LOCKSS installation process
```

and **install-lockss** will terminate.

### 3.3.11 Checking the K3s Configuration

**Tip:** This section is optional.

K3s comes with **k3s check-config**, a configuration checker tool. The K3s configuration checker is capable of detecting complex underlying system situations that definitely require fixing (or applications running in the K3s cluster will not be able to function properly). On the other hand, the versions of the K3s configuration checker available at the time LOCKSS 2.0-alpha7 was released contained bugs that reported spurious issues that are either inaccurate or moot. As a result, we have decided against running **k3s check-config** as part of **install-lockss** at this time, to avoid

unnecessary interruptions in the installation of the LOCKSS system in many cases where there is no particular cause for concern.

That being said, we still recommend running **k3s check-config** and interpreting the results using the *Troubleshooting the K3s Configuration Checker* section of the manual:

1. Run this command:

```
k3s check-config
```

2. The following error messages in the output are indicative of system situations that require attention:

```
/usr/sbin iptables v1.8.2 (nf_tables): should be older than v1.8.0, newer than v1.8.
↪3, or in legacy mode (fail)
```

```
RHEL7/CentOS7: User namespaces disabled; add 'user_namespace.enable=1' to boot_
↪command line (fail)
```

```
apparmor: enabled, but apparmor_parser missing (fail)
```

---

### Troubleshooting

See *Troubleshooting the K3s Configuration Checker* for details.

---

3. The following error messages in the output can be ignored:

```
cgroup hierarchy: nonexistent?? (fail)
```

```
links: aux/ip6tables should link to iptables-detect.sh (fail)
links: aux/ip6tables-restore should link to iptables-detect.sh (fail)
links: aux/ip6tables-save should link to iptables-detect.sh (fail)
links: aux/iptables should link to iptables-detect.sh (fail)
links: aux/iptables-restore should link to iptables-detect.sh (fail)
links: aux/iptables-save should link to iptables-detect.sh (fail)
```

```
swap: should be disabled
```

```
CONFIG_INET_XFRM_MODE_TRANSPORT: missing
```

---

### Troubleshooting

See *Troubleshooting the K3s Configuration Checker* for details.

---

4. For other error messages, check the official [K3s documentation](#), search for [K3s issues database on GitHub](#) or the Web for resources matching your error message or operating system, and/or contact us so we can help investigate and document for future reference.
-

## CONFIGURING THE LOCKSS SYSTEM

After [installing the LOCKSS system](#), you will configure it with the **configure-lockss** script. If you have experience with classic LOCKSS daemon version 1.x, this is the equivalent of **hostconfig**.

### 4.1 Before Invoking **configure-lockss**

You will need to gather information to answer configuration questions asked by **configure-lockss**, including:

- The name (FQDN) of the host, the IP address of the host, and if behind NAT, the external IP address for NAT.
- The mail relay host, and optionally mail credentials, for sending e-mail from the host, and the e-mail address for the administrator of the system.
- The configuration URL and preservation group or groups corresponding to the LOCKSS network your system is joining.
- The paths for the primary content storage area, any additional content storage areas, the state data storage area, the temporary storage area, and the log storage area. See the [Storage](#) and [Network-Attached Storage](#) sections for important information about performance requirements for these storage areas.

**Caution:** Each of these paths needs to be writeable by the lockss user. If this is not the case, set them up as root before running **configure-lockss**.

- Username and password for the Web user interfaces.
- A password for the PostgreSQL database.
  - Alternatively, if using an existing PostgreSQL database, the host name, port, schema, username and password for the external PostgreSQL database, as well as a prefix for database names.
- A username and password for the Solr database.
  - Alternatively, if using an existing Solr database, the host name, port, username and password for the external Solr database, as well as the core name for the LOCKSS repository.
- Whether you wish to use the LOCKSS Crawler Service, LOCKSS Metadata Extraction Service, LOCKSS Metadata Service, LOCKSS SOAP Compatibility Service, Pywb Web replay engine, and OpenWayback Web replay engine.

Some notes about using **configure-lockss**:

- When run the first time, some of the questions asked by the script will have a suggested or default 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 as the default value; review and hit **Enter** to leave unchanged.

- Password prompts will not display the previous value but can still be left unchanged with Enter.

## 4.2 Invoking `configure-lockss`

To invoke **`configure-lockss`**, simply run this command in the lockss user's `lockss-installer` directory as lockss<sup>1</sup>:

`scripts/configure-lockss`

The script will begin with the first series of configuration questions, about *Kubernetes Settings*.

## 4.3 Kubernetes Settings

Prompt: *Command to use to execute kubectl commands*

Enter the command to invoke **`kubectl`** in your environment. If you are using the K3s Kubernetes environment that ships with the LOCKSS system, the proposed value is already correct.

## 4.4 Network Settings

### 4.4.1 Hostname

Prompt: *Fully qualified hostname (FQDN) of this machine*

Enter the machine's fully-qualified hostname (meaning with its domain name), for example `locksstest.myuniversity.edu`.

### 4.4.2 IP Address

Prompt: *IP address of this machine*

If the machine is publicly routable, meaning it has an IP address that can be used to identify it over the Internet, enter the publicly routable IP address. Otherwise, if the machine is accessible via network address translation (NAT), meaning it has an IP address that is valid only on your local network but it can be reached from the Internet via a NAT router, enter the internal IP address.

### 4.4.3 Network Address Translation

1. Prompt: *Is this machine behind NAT?*

If the machine is publicly routable, enter N; otherwise, if the machine is not publicly routable but will be accessible via network address translation (NAT), enter Y.

2. If you answered Y, you will be asked an additional configuration question:

*External IP address for NAT*

Enter the publicly routable IP address of the NAT router.

---

<sup>1</sup> See *Running Commands as the lockss User*.



#### 4.4.4 Initial UI Subnet

Prompt: *Initial subnet(s) for admin UI access*

Enter a semicolon-separated list of subnets in CIDR or mask notation that should initially have access to the Web user interfaces (UI) of the system. The access list can be modified later via the UI.

#### 4.4.5 Container Subnet

1. If **configure-lockss** detects a discrepancy between a previously used subnet for inter-container communication in the system and the subnet it would choose now, you may either see the warning:

*Container subnet has changed from <former\_subnet> to <new\_subnet>*

or be asked the question:

*Container subnet was <former\_subnet>, we think it should now be <new\_subnet>. Do you want to change it?*  
in which case you should enter Y (recommended) or N.

2. Prompt: *LOCKSS subnet for inter-service access control*

Enter the subnet used for inter-container communication. We recommend accepting the proposed value by hitting Enter.

#### 4.4.6 LCAP Port

Prompt: *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.

### 4.5 Mail Settings

#### 4.5.1 Mail Relay

Prompt: *Mail relay for this machine*

Enter the hostname of this machine's outgoing mail server, for example `smtp.myuniversity.edu`.

#### 4.5.2 Mail Relay Credentials

1. Prompt: *Does the mail relay <mailhost> need a username and password?*

If the outgoing mail server does not require password authentication, enter N; otherwise, enter Y.

2. If you answered Y, you will be asked additional configuration questions:

1. Prompt: *User for <mailhost>*

Enter a username for the mail server.

2. Prompt: *Password for <mailuser>@<mailhost>*

Enter the password for the username on the mail server.

3. Prompt: *Password for <mailuser>@<mailhost> (again)*

Re-enter the password for the username on the mail server. If the two passwords do not match, the password will be asked again.

### 4.5.3 Administrator Email

Prompt: *E-mail address for administrator*

Enter the e-mail address of the person or team who will administer the LOCKSS system on this machine.

## 4.6 Preservation Network Settings

### 4.6.1 Configuration URL

1. Prompt: *Configuration URL*

Enter the URL of your LOCKSS network's configuration file. Select a scenario below for more details:

#### LOCKSS Demo Network

If you are trying out LOCKSS 2.x, enter `http://props.lockss.org:8001/demo/lockss.xml` (or simply hit **Enter**, as this is the default).

#### Global LOCKSS Network

If you are participating in the Global LOCKSS Network and trying out LOCKSS 2.x, enter `http://props.lockss.org:8001/demo/lockss.xml` (or simply hit **Enter**, as this is the default).

#### Other LOCKSS Network

If you are configuring your LOCKSS node to participate in a given LOCKSS network, enter the configuration URL provided for that LOCKSS network by your administrators (for example `https://admin.mynetwork.org/config/lockss.xml`).

2. If the configuration URL begins with `https:`, you will be asked additional configuration questions:

1. Prompt: *Verify configuration server authenticity?*

Enter **Y** if you would like to check the authenticity of the configuration server using a custom keystore; otherwise enter **N**.

2. If you answered **Y**, you will be asked an additional configuration question:

*Server certificate keystore*

Enter the path of a Java keystore used to verify the authenticity of the configuration server.

### 4.6.2 Configuration Proxy

Prompt: *Configuration proxy (host:port)*

If the configuration URL can be reached directly, leave this blank; otherwise, if a proxy server is required to reach the configuration URL, enter its host and port in *host:port* format (for example *proxy.myuniversity.edu:8080*).

### 4.6.3 Preservation Groups

Prompt: *Preservation group(s)*

Enter a preservation group identifier or semicolon-separated list of preservation group identifiers. Select a scenario below for more details:

#### **LOCKSS Demo Network**

If you are trying out LOCKSS 2.x, enter *demo* (or simply hit **Enter**, as this is the default).

#### **Global LOCKSS Network**

If you are participating in the Global LOCKSS Network and trying out LOCKSS 2.x, enter *demoprod*.

#### **Other LOCKSS Network**

If you are configuring your LOCKSS node to participate in a given LOCKSS network, enter the preservation group(s) provided for that LOCKSS network by your administrators (for example *mynetwork*, or *mynetwork;mygroup1;mygroup2*).

## 4.7 Storage Areas

The LOCKSS system needs several kinds of storage areas, as described in the [Storage](#) section. See also the [Network-Attached Storage](#) section for important information about performance requirements for these storage areas.

Depending on your host system's layout, these storage areas may all be the same, or all be different mount points or paths. Each path must be writeable by the *lockss* user.

Subdirectories will be created in each storage area to fit the needs of each system component; for example *lockss-stack-cfg-data* is the LOCKSS configuration service's state data directory in the state data storage area, and *lockss-stack-repo-logs* is the LOCKSS repository service's log directory in the log storage area.

### 4.7.1 State Data Storage Area

Prompt: *Root path for state data storage*

This directory is used as the root of the storage area for databases and other state data. Enter the desired path, or if reconfiguring, hit **Enter** to accept a previously-entered value.

## 4.7.2 Content Storage Areas

1. Prompt: *Root path(s) for content storage*

Enter a semicolon-separated list of full paths of directories to be used to store preserved content.

2. If the answer to the question is different than that from a previous configuration run, you will see the warning:

If you have removed or reordered content storage directories, you must run `scripts/reindex-artifacts`

If you have done anything other add new content storage areas to the end of the previously-entered list, you must run `scripts/reindex-artifacts` after completion of **configure-lockss**, before starting the system.

## 4.7.3 Log Storage Area

Prompt: *Root path for log storage*

This directory is used as the root of the storage area for log files in the LOCKSS system. Accept the default (same directory as the content data storage directory root) by hitting **Enter**, or enter a custom path.

## 4.7.4 Temporary Storage Area

Prompt: *Root path for temporary storage (local storage preferred)*

This directory is used as the root of the storage area for temporary files in the LOCKSS system. Accept the default (same directory as the content data storage directory root) by hitting **Enter**, or enter a custom path.

## 4.8 Web User Interface Settings

1. Prompt: *User name for web UI administration*

Enter a username for the primary administrative user in the LOCKSS system's Web user interfaces.

2. Prompt: *Password for web UI administration user <uiuser>*

Enter a password for the primary administrative user.

3. Prompt: *Password for web UI administration user <uiuser> (again)*

Re-enter the password for the primary administrative user. If the two passwords do not match, the password will be asked again.

## 4.9 Database Settings

### 4.9.1 PostgreSQL

Prompt: *Use embedded LOCKSS PostgreSQL DB Service?*

Select **either** option A **or** option B:

- A. Enter Y to use the **embedded PostgreSQL database**. This is recommended in most cases; a PostgreSQL database will be run and managed by the LOCKSS system internally. If you choose this option, see [Embedded PostgreSQL Database](#).

- B. Enter N to use an **external PostgreSQL database**. Select this option if you wish to use an existing PostgreSQL database at your institution or one that you run and manage yourself. If you choose this option, see [External PostgreSQL Database](#).

### Embedded PostgreSQL Database

If you select this option, you will be asked additional configuration questions:

1. Prompt: *Password for PostgreSQL database*

Enter the password for the embedded PostgreSQL database.

**Warning:** This prompt is used to record the PostgreSQL database password in the LOCKSS system's configuration. If you change the value of the PostgreSQL database password here without actually changing the PostgreSQL database password, the LOCKSS system components will no longer be able to connect to the PostgreSQL database. See [Working with PostgreSQL](#) for details.

2. Prompt: *Password for PostgreSQL database (again)*

Re-enter the password for the embedded PostgreSQL database. If the two passwords do not match, the password will be asked again.

3. Complete the [Solr](#) section next.

### External PostgreSQL Database

If you select this option, you will be asked additional configuration questions:

1. Prompt: *Fully qualified hostname (FQDN) of PostgreSQL host*

Enter the hostname of the external PostgreSQL database, for example `postgres.myuniversity.edu`.

2. Prompt: *Port used by PostgreSQL host*

Enter the port where the external PostgreSQL database can be reached, for example 5432.

3. Prompt: *Schema for PostgreSQL service*

Enter the schema name to be used by the LOCKSS system. The schema name used in the embedded PostgreSQL database is LOCKSS, but your database administrator may assign a different schema name to you.

4. Prompt: *Database name prefix for PostgreSQL service*

Enter the prefix to use for any LOCKSS-related database names in the schema. The database name prefix in the embedded PostgreSQL database is Lockss (note the uppercase/lowercase), but your database administrator may assign a different database name prefix.

5. Prompt: *Login name for PostgreSQL service*

Enter the username for the external PostgreSQL database. The username in the embedded PostgreSQL database is LOCKSS, but your database administrator may assign a different username to you.

6. Prompt: *Password for PostgreSQL database*

Enter the password for the username in the external PostgreSQL database.

**Warning:** This prompt is used to record the PostgreSQL database password in the LOCKSS system's configuration. If you change the value of the PostgreSQL database password here without actually changing the PostgreSQL database password, the LOCKSS system components will no longer be able to connect to the PostgreSQL database. Contact your PostgreSQL database administrator for details.

7. Prompt: *Password for PostgreSQL database (again)*

Re-enter the password for the username in the external PostgreSQL database. If the two passwords do not match, the password will be asked again.

8. Complete the *Solr* section next.

### 4.9.2 Solr

Prompt: *Use embedded LOCKSS Solr Service?*

Select **either** option A **or** option B:

- A. Enter Y to use the **embedded Solr database**. This is recommended in most cases; a Solr database will be run and managed by the LOCKSS system internally. If you choose this option, see [Embedded Solr Database](#).
- B. Enter N to use an **external Solr database**. Select this option if you wish to use an existing Solr database at your institution or one that you run and manage yourself. If you choose this option, see [External Solr Database](#).

#### Embedded Solr Database

If you select this option, you will be asked additional configuration questions:

1. Prompt: *User name for LOCKSS Solr access*

Enter the username for the embedded Solr database.

2. Prompt: *Password for LOCKSS Solr access*

Enter the password for the username in the embedded Solr database.

3. Prompt: *Password for LOCKSS Solr access (again)*

Re-enter the password for the username in the embedded Solr database. If the two passwords do not match, the password will be asked again.

4. Complete the *Metadata Query Service* section next.

#### External Solr Database

If you select this option, you will be asked additional configuration questions:

1. Prompt: *Fully qualified hostname (FQDN) of Solr host*

Enter the hostname of the external Solr database server, for example `solr.myuniversity.edu`.

2. Prompt: *Port used by Solr host:*

Enter the port used by the database server on the Solr host, for example 8983.

3. Prompt: *Solr core repo name:*

Enter name of the Solr core for the LOCKSS repository. The Solr core name used in the embedded Solr database is `lockss-repo`, but your database administrator may assign a different Solr core name.

4. Prompt: *User name for LOCKSS Solr access*

Enter the username for the external Solr database.

5. Prompt: *Password for LOCKSS Solr access*

Enter the password for the username in the external Solr database.

6. Prompt: *Password for LOCKSS Solr access (again)*

Re-enter the password for the username in the external Solr database. If the two passwords do not match, the password will be asked again.

7. Complete the *Metadata Query Service* section next.

## 4.10 LOCKSS Services

### 4.10.1 Crawler Service

1. Prompt: *Use LOCKSS Crawler Service?*

Enter Y if you want the crawler service to be run, otherwise N. (The only situation where a crawler service is not needed is LOCKSS networks that are exclusively using direct deposit to store content, most LOCKSS networks need the crawler service.)

2. If you answer Y: to the previous question, you will see these additional questions:

1. Prompt: *Enable classic LOCKSS crawler?*

Enter Y if you want to run the classic LOCKSS crawler, otherwise N. (Most LOCKSS networks using the crawler service use the classic LOCKSS crawler.)

2. Prompt: *Enable Wget crawler?*

Enter Y if you want to enable the usage of the external Wget crawler, otherwise N.

### 4.10.2 Metadata Query Service

Prompt: *Use LOCKSS Metadata Query Service?*

Enter Y if you want the metadata query service to be run, otherwise N.

### 4.10.3 Metadata Extraction Service

Prompt: *Use LOCKSS Metadata Extraction Service?*

Enter Y if you want the metadata extraction service to be run, otherwise N.

## 4.10.4 SOAP Compatibility Service

Prompt: *Use LOCKSS SOAP Compatibility Service?*

Enter Y if you want the SOAP compatibility service to be run, otherwise N. (This is only needed if you have external tools using the LOCKSS' legacy SOAP Web Services.)

## 4.11 Web Replay Settings

### 4.11.1 Pywb

Prompt: *Use LOCKSS Pywb Service?*

Enter Y to run an embedded Pywb engine for Web replay; otherwise, enter N.

### 4.11.2 OpenWayback

1. Prompt: *Use LOCKSS OpenWayback Service?*

Enter Y to use an embedded OpenWayback engine for Web replay; otherwise, enter N.

2. If you answered Y, you will be asked an additional configuration question:

*Okay to turn off authentication for read-only requests for LOCKSS Repository Service?*

OpenWayback currently does not supply user credentials when reading content from the LOCKSS repository, so the repository must be configured to respond to unauthenticated read requests. Enter Y to accept this, otherwise you will see the warning:

*Not enabling OpenWayback Service*

and OpenWayback will not be run.

## 4.12 Final Steps

1. Prompt: *OK to store this configuration?*

Enter Y if the configuration values are to your liking; otherwise, enter N to make edits.

2. If you answer Y, **configure-lockss** will perform the final configuration steps. You may be asked to confirm before directories are created for the first time:

*<directory> does not exist; shall I create it?*

or before directory permissions are changed:

*<directory> is not writable; shall I chown it?*

In each case, enter Y for "yes" and N for "no".

---



## RUNNING THE LOCKSS SYSTEM

The commands in this section are all run as the `lockss` user<sup>1</sup> in the `lockss` user's `lockss-installer` directory.

### 5.1 Starting the LOCKSS System

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

- `scripts/generate-lockss`: This script takes your configuration data and turns it 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 Kubernetes.

The **`start-lockss`** accepts some options:

**`--update (-u)`**

Force the system to check for newer container images of the system's components (LOCKSS services, embedded databases, embedded Web replay engines...) before deploying the system to Kubernetes.

**`--wait (-w)`**

After deploying the system to Kubernetes and waiting for the system's containers to come up, additionally wait for an internal signal from the system that the system's components are fully initialized. (Currently this internal signal comes from the poller service.)

### 5.2 Shutting down the LOCKSS System

Run `scripts/stop-lockss`.

---

<sup>1</sup> See *Running Commands as the lockss User*

## 5.3 Restarting a Running LOCKSS System

Run `scripts/restart-lockss`.

The **restart-lockss** accepts the same options as **start-lockss**.

## 5.4 Removing a Configured LOCKSS System

To remove all configurations, volumes and networks configured by the LOCKSS system in Kubernetes, run `scripts/uninstall-lockss`. This will **not** remove files from the persistent store.

---

## USING THE LOCKSS SYSTEM

This chapter describes how to use the LOCKSS system.

These sections are under construction. The user interface of the LOCKSS 2.x is currently largely similar to that of the classic LOCKSS system (1.x); you can refer to the classic LOCKSS system manual in the meantime for similar information.

### 6.1 Using the LOCKSS Configuration Service

---

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

---

#### 6.1.1 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.

#### 6.1.2 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.

### 6.1.3 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.

### 6.1.4 Managing Access to the Web User Interfaces

*This section is under construction.*

## 6.2 Using the LOCKSS Crawler Service

---

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

---

### 6.2.1 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.

### 6.2.2 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.

### 6.2.3 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.

### 6.2.4 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.

## 6.3 Using the LOCKSS Poller Service

---

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

---

### 6.3.1 Accessing the Web User Interface

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *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.

### 6.3.2 Requesting Polls

*This section is under construction.*

### 6.3.3 Monitoring Polling and Voting

*This section is under construction.*

## 6.4 Using the LOCKSS Metadata Extraction Service

---

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

---

### 6.4.1 Accessing the Web User Interface

If you are already connected to the Web user interface (UI) of another component of the LOCKSS System, click *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.

### 6.4.2 Requesting Metadata Extraction

*This section is under construction.*

## 6.5 Using the LOCKSS Metadata Service

---

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

---

### 6.5.1 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.

### 6.5.2 Requesting Metadata Information

*This section is under construction.*

## 6.6 Replaying Web Content with Pywb

### 6.6.1 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`.

### 6.6.2 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.

### 6.6.3 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.

## 6.7 Replaying Web Content with OpenWayback

### 6.7.1 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`.

### 6.7.2 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*.

### 6.7.3 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.

## 6.8 Using the Kubernetes Dashboard

Kubernetes comes with the [Kubernetes Dashboard](#), a web-based user interface (UI).

To facilitate installing and interacting with the Kubernetes Dashboard, the LOCKSS Installer offers the `dashboard-util` script.

### 6.8.1 Installing the Kubernetes Dashboard

To install the Kubernetes Dashboard, run this command<sup>1</sup>:

```
scripts/dashboard-util --install
```

If the installation succeeds, the program will also display the login URL and the bearer token.

### 6.8.2 Accessing the Kubernetes Dashboard

To access the Kubernetes Dashboard:

1. Create a secure channel to your K3s cluster with the following command:

```
k3s kubectl proxy &
```

**Note:** This command runs in the background "forever".

2. Obtain the login URL with the following command:

```
scripts/dashboard-util --url
```

3. Obtain the login token with the following command:

```
scripts/dashboard-util --token
```

4. Open a browser and go to the login URL.
5. Make sure the *Token* radio button is selected.
6. Copy and paste the login token into the *Enter token* text field.

---

<sup>1</sup> This command is relative to the lockss user's `lockss-installer` directory.

### 6.8.3 Using the Kubernetes Dashboard UI

When the dashboard comes up, it will be in the default namespace. Click on the namespace pull-down menu near the top and select the `lockss` namespace to see the LOCKSS components. If all of your deployments are running and ready, the three circles at the top should be green. In the left hand panel you can select the components you are interested in:

- Click on *Services* to see the cluster IP for each of the running services. You can click on a specific service to see more detailed information.
- Click on *Deployments* to see a list of services and their CPU and memory usage. You can access specific services and deployments from here.
- Click on *Pods*. This will give you information about all the pods running. Click on a pod of interest to obtain more granular information:

#### *View logs*

Since LOCKSS output logs are persisted to a local directory, there will be very little in the Kubernetes logs if the container came up without errors.

#### *Exec into pods*

This will open a terminal window into the container.

#### *Edit the pod resource*

This will allow you to view and edit the YAML file which was used to start the pod. The edit will not persist on restart.

#### *Delete the pod*

While this will delete the current pod, a new pod will be spawned by the deployment with a new pod ID.

### 6.8.4 Updating the Kubernetes Dashboard

To update the Kubernetes Dashboard to the most recent release, run this command<sup>Page 45, 1</sup>:

```
scripts/dashboard-util --update
```

### 6.8.5 Removing the Kubernetes Dashboard

To remove the Kubernetes Dashboard from the `kubernetes-dashboard` namespace, run this command<sup>Page 45, 1</sup>:

```
scripts/dashboard-util --remove
```

---

#### See Also

- [Web UI \(Dashboard\)](#) on the Kubernetes website.
-

## TROUBLESHOOTING THE LOCKSS SYSTEM

This chapter contains sections of additional information about troubleshooting the LOCKSS system.

Sections include guidance for troubleshooting networking-related components like **iptables**, **firewalld**, and **ufw**, and K3s-related components like CoreDNS, the K3s Installer, the K3s Configuration Checker.

### 7.1 Known Issues

*This section was last updated: 2023-01-23.*

#### Expired K3s Certificate

LOCKSS-provided scripts like `scripts/stop-lockss` or `scripts/restart-lockss` that interact with the K3s server may display error messages that include:

```
error: You must be logged in to the server (Unauthorized)
```

In parallel, the host system log might contain messages similar to:

```
client_builder_dynamic.go:197: get or create service account failed: Get "https://127.0.
↪0.1:6444/api/v1/namespaces/kube-system/serviceaccounts/generic-garbage-collector":_
↪x509: certificate has expired or is not yet valid: current time 2023-01-22T03:32:05-
↪08:00 is after 2023-01-21T02:13:28Z
```

This is caused by a bug in the version of K3s that currently ships with the LOCKSS system, whereby the auto-rotation of a certificate after the K3s server runs continuously for one year does not take place correctly. Future releases of the LOCKSS system will use a newer version of K3s without this bug. In the meantime, to work around this problem, simply restart the K3s service with **systemctl** as **root**:

```
systemctl restart k3s
```

Scripts that interact with the K3s server should then again work as expected.

## Security

- In the "alpha" phase of development of LOCKSS 2.0, there are no access controls on Kubernetes' API. It is not accessible from outside the machine, but any local user can access the API, so they can stop the LOCKSS containers, change their contents, read secrets, etc. We plan to enable access controls in the "beta" phase.
- In the Classic LOCKSS system (1.x), the LCAP SSL key could only be read by `root`, but now it can also be read by `lockss`.

## DNS Resolution

K3s' default DNS cache timeout is 30 seconds, which results in enough repetitive upstream queries to trigger alarms at some institutions. One remediation is to change the CoreDNS configuration by editing its configmap.

With K3s, changes made to CoreDNS's configmap with **kubect1 apply** do not persist, because the configmap is constantly reloaded from `/var/lib/rancher/k3s/server/manifests/coredns.yaml`. Additionally, K3s overwrites the file with the defaults at startup, so changes there are not really persistent either.

The LOCKSS Installer offers the script `scripts/coredns-cron-hack`, which sets the CoreDNS cache timeout to 30 minutes. It should be run once, as `root`, after each time K3s starts. Absent a good way to do that, it is harmless to run it periodically from `root`'s crontab. The recommended use is to copy it to a `root`-owned file in `/etc/cron.hourly`.

## Harmless PID File Errors

The stdout log files of the various LOCKSS service containers contain the following error messages at startup:

```
/usr/local/bin/docker-entrypoint: line 374: can't create /var/run/docker-entrypoint.pid:
↪Permission denied
```

This is harmless and will be addressed in the a future release of the system.

## 7.2 Troubleshooting iptables

This section provides troubleshooting information for the *Configuring iptables for K3s* phase of `/installing/installer`.

### 7.2.1 Switch iptables to legacy mode via Alternatives

K3s 1.21.5+k3s1 (the version used by LOCKSS 2.0-alpha7) does not always work with **iptables** versions 1.8.0-1.8.3 when run via Alternatives but not in legacy mode, for instance in some Debian or Ubuntu systems<sup>1</sup>. If **install-lockss** detects this situation, you will see a warning message and the following prompt<sup>2</sup>:

*Switch iptables to legacy mode via Alternatives?*

---

<sup>1</sup> References:

- <https://rancher.com/docs/k3s/latest/en/known-issues/>
- <https://github.com/kubernetes/kubernetes/issues/71305>
- <https://github.com/k3s-io/k3s/issues/116>
  - <https://github.com/k3s-io/k3s/issues/116#issuecomment-624770403>
- <https://github.com/k3s-io/k3s/issues/703>

<sup>2</sup> See *Configuring iptables for K3s*.

Enter Y to accept the proposed **iptables** configuration. **If you bypass the proposed configuration, K3s may malfunction.**

The remediation attempted by **install-lockss** is equivalent to:

```
# Required only if ufw is active
ufw disable

# Required
update-alternatives --set iptables /usr/sbin/iptables-legacy

# Required
update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy

# Optional
update-alternatives --set arptables /usr/sbin/arptables-legacy

# Optional
update-alternatives --set ebtables /usr/sbin/ebtables-legacy

# Required only if ufw was active
ufw enable
```

## 7.2.2 Post-Installation Changes to iptables

If your system did not initially need an adjustment for **iptables** at the time K3s was installed, but later does (for example because **iptables** is upgraded from a pre-1.8.0 version to version 1.8.0 or later), run this command (relative to the *LOCKSS Installer Directory*) as a privileged user who can become root via **sudo**<sup>3</sup>:

```
scripts/install-lockss --configure-iptables
```

This will run only the *Configuring iptables for K3s* phase of **install-lockss**.

## 7.3 Troubleshooting firewalld

This section provides troubleshooting information for the *Configuring firewalld for K3s* phase of /installing/installer.

### 7.3.1 Add 10.42.0.0/16 and 10.43.0.0/16 to firewalld's trusted zone

If your system is running the **firewalld** firewall, it is necessary to add K3s' pod and service subnets<sup>1</sup> to **firewalld**'s trusted zone for K3s to work properly<sup>2</sup>. If **install-lockss** detects this situation, you will see a warning message

<sup>3</sup> See *Running Commands as a Privileged User*.

<sup>1</sup> By default, K3s' pod subnet is 10.42.0.0/16 and service subnet is 10.43.0.0/16.

<sup>2</sup> For operating systems in the RHEL family (CentOS, Rocky Linux, AlmaLinux OS...), the action recommended by the K3s manual is to disable **firewalld** entirely (see <https://rancher.com/docs/k3s/latest/en/advanced/#additional-preparation-for-red-hat-centos-enterprise-linux>), but **install-lockss** takes a lighter approach commonly used in the K3s community.

References:

- <https://github.com/k3s-io/k3s/issues/1556>
  - <https://github.com/k3s-io/k3s/issues/1556#issuecomment-604112415>

and the following prompt<sup>3</sup>:

*Add 10.42.0.0/16 and 10.43.0.0/16 to firewalld's trusted zone?*

Enter Y to accept the proposed **firewalld** configuration. **If you bypass the proposed configuration, K3s may malfunction.**

The **firewalld** configuration attempted by **install-lockss** is equivalent to<sup>Page 49, 1</sup>:

```
firewall-cmd --permanent --zone=trusted --add-source=10.42.0.0/16
firewall-cmd --permanent --zone=trusted --add-source=10.43.0.0/16
firewall-cmd --reload
```

### 7.3.2 Post-Installation Changes to firewalld

If your system did not initially use **firewalld** at the time K3s was installed, but later does (for example because **firewalld** becomes enabled), run this command (relative to the *LOCKSS Installer Directory*) as a privileged user who can become root via **sudo**<sup>4</sup>:

```
scripts/install-lockss --configure-firewalld
```

This will run only the *Configuring firewalld for K3s* phase of **install-lockss**.

---

## 7.4 Troubleshooting ufw

This section provides troubleshooting information for the *Configuring ufw for K3s* phase of /installing/installer.

### 7.4.1 Allow traffic from 10.42.0.0/16 and 10.43.0.0/16 via ufw

If your system is running the **ufw** firewall, it is necessary to allow traffic from K3s' pod and service subnets<sup>1</sup> via **ufw** for K3s to work properly<sup>2</sup>. If **install-lockss** detects this situation, you will see a warning message and the following prompt<sup>3</sup>:

*Allow traffic from 10.42.0.0/16 and 10.43.0.0/16 via ufw?*

Enter Y to accept the proposed **ufw** configuration. **If you bypass the proposed configuration, K3s may malfunction.**

The **firewalld** configuration attempted by **install-lockss** is equivalent to<sup>1</sup>:

---

<sup>3</sup> See *Configuring firewalld for K3s*.

<sup>4</sup> See *Running Commands as a Privileged User*.

<sup>1</sup> By default, K3s' pod subnet is 10.42.0.0/16 and service subnet is 10.43.0.0/16.

<sup>2</sup> References:

- <https://github.com/k3s-io/k3s/issues/1280>
  - <https://github.com/k3s-io/k3s/issues/1280#issuecomment-663269728>

<sup>3</sup> See *Configuring ufw for K3s*.

```
ufw allow from 10.42.0.0/16 to any
ufw allow from 10.43.0.0/16 to any
ufw reload
```

## 7.4.2 Post-Installation Changes to ufw

If your system did not initially use **ufw** at the time K3s was installed, but later does (for example because **ufw** becomes enabled), run this command (which is relative to the *LOCKSS Installer Directory*) as a privileged user who can become root via **sudo**<sup>4</sup>:

```
scripts/install-lockss --configure-ufw
```

This will run only the *Configuring ufw for K3s* phase of **install-lockss**.

## 7.5 Troubleshooting CoreDNS

This section provides troubleshooting information for the *Configuring CoreDNS for K3s* phase of /installing/installer.

### 7.5.1 CoreDNS does not allow a loopback address to be given to Kubernetes pods as an upstream DNS server

If both `/etc/resolv.conf` and `/run/systemd/resolve/resolv.conf` (files used to list the IP address of DNS servers) contain loopback addresses, CoreDNS (a component of the K3s Kubernetes cluster that handles DNS resolution) will not work properly<sup>1</sup>. If **install-lockss** detects this situation, you will see a warning message including CoreDNS does not allow a loopback address to be given to Kubernetes pods as an upstream DNS server, and the following prompt<sup>2</sup>:

*IP address(es) of DNS resolvers, separated by ';'.*

Enter a semicolon-separated list of DNS server IP addresses that are *not* loopback addresses or hit Enter to accept the proposed value.

### 7.5.2 Post-Installation Changes to DNS

If the DNS settings of your system change after K3s is initially installed (for example if DNS servers are added or removed), run this command (which is relative to the *LOCKSS Installer Directory*) as a privileged user who can become root via **sudo**<sup>3</sup>:

```
scripts/install-lockss --configure-coredns
```

<sup>4</sup> See *Running Commands as a Privileged User*.

<sup>1</sup> References:

- <https://coredns.io/plugins/loop/#troubleshooting-loops-in-kubernetes-clusters>

<sup>2</sup> See *Configuring CoreDNS for K3s*.

<sup>3</sup> See *Running Commands as a Privileged User*.

This will run only the *Configuring CoreDNS for K3s* phase of **install-lockss**.

---

## 7.6 Troubleshooting the K3s Installer

The LOCKSS Installer's **install-lockss** script installs K3s by executing Rancher's official K3s Installer from <https://get.k3s.io/>, after checking that various system, firewall and DNS prerequisites are addressed (see *installing/installer*). However, the installation can still run into issues and fail. Some of the error messages you might encounter are documented below, but you may need to refer to the official [K3s documentation](#) or use a search engine to look up a specific error message.

### 7.6.1 Enabling User Namespaces in RHEL 7 and CentOS 7

K3s requires user namespaces, a feature generally available and enabled in many Linux flavors. However, some RHEL 7 and CentOS 7 systems do not have user namespace enabled by default. This can cause the *Checking K3s Prerequisites* or *Testing the K3s Node* phases of **install-lockss** or the optional *Checking the K3s Configuration* phase to fail.

To resolve this issue in RHEL 7 or CentOS 7<sup>1</sup>:

1. Edit the file `/etc/default/grub` as root<sup>2</sup>.

1. Look for the line beginning with `GRUB_CMDLINE_LINUX=`, for example:

```
GRUB_CMDLINE_LINUX="no_timer_check console=tty0 console=ttyS0,115200n8 net.  
↪ifnames=0 biosdevname=0 elevator=noop crashkernel=auto"
```

This line defines a space-separated list of boot arguments.

2. Add `user_namespace.enable=1` to the beginning of the space-separated list, for instance:

```
GRUB_CMDLINE_LINUX="user_namespace.enable=1 no_timer_check console=tty0  
↪console=ttyS0,115200n8 net.ifnames=0 biosdevname=0 elevator=noop  
↪crashkernel=auto"
```

2. Run the following command as root:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

---

**Note:** If EFI is in use, the file to use for the `-o` option will not be `/boot/grub2/grub.cfg` exactly. For example on CentOS, it may be `/boot/efi/EFI/centos/grub.cfg`.

---

3. Reboot the system.
4. Check that the change took effect. To do so, look at the contents of `/proc/cmdline` (for example type `cat /proc/cmdline`) and verify that it now contains `user_namespace.enable=1`.

---

**Note:** If the change did not take effect, it could be an indication that a different `grub.cfg` file is needed for the `-o` option of **grub2-mkconfig** command above.

---

---

<sup>1</sup> References:

- <https://fortuitousengineer.com/installing-kubernetes-k3s-on-centos-rhel-hosts/>

<sup>2</sup> See *Running Commands as root*.



## 7.6.2 Installing apparmor\_parser

K3s uses Apparmor in systems where it is enabled. However, some systems, especially OpenSUSE systems, have Apparmor enabled but **apparmor\_parser** is not installed by default. This can cause the *Checking K3s Prerequisites* or *Installing K3s* phases of **install-lockss** or the optional *Checking the K3s Configuration* phase to fail.

To resolve this issue in OpenSUSE, run these **zypper** commands as **root** <sup>Page 52, 2</sup>:

```
zypper refresh

zypper --non-interactive install apparmor-parser
```

or equivalently:

```
zypper refresh

zypper -n install apparmor-parser
```

**Tip:** In other Linux flavors, use similar package installation commands.

## 7.6.3 Failed to apply container\_runtime\_exec\_t to /usr/local/bin/k3s

In some Fedora systems, the K3s installer may fail with an error message similar to the following:

```
[ERROR] Failed to apply container_runtime_exec_t to /usr/local/bin/k3s, please install:
    yum install -y container-selinux selinux-policy-base
    yum install -y https://rpm.rancher.io/k3s/stable/common/centos/8/noarch/k3s-selinux-
    ↪0.3-0.el8.noarch.rpm
```

The specific commands and version numbers may vary from the example above.

To resolve this problem, run the recommended commands as **root** <sup>Page 52, 2</sup>.

## 7.6.4 k3s-selinux requires container-selinux

In some Oracle Linux 7 systems, you may see an error message similar to the following:

```
Error: Package: k3s-selinux-0.3-0.el7.noarch (rancher-k3s-common-stable)
    Requires: container-selinux >= 2.107-3
You could try using --skip-broken to work around the problem
You could try running: rpm -Va --nofiles --nodigest
```

The specific commands and version numbers may vary from the example above.

This can occur in environments where the Oracle Linux 7 Addons Yum repository is not enabled by default, so Rancher's official K3s installer is unable to install the package **container-selinux** automatically.

To resolve this problem in Oracle Linux 7, run the following command as **root** <sup>Page 52, 2</sup>:

```
yum-config-manager --enable ol7_addons
```

## 7.7 Troubleshooting the K3s Configuration Checker

After installing K3s<sup>2</sup>, you may optionally run the K3s configuration checker **k3s check-config**<sup>3</sup> (see *Checking the K3s Configuration*). This configuration checker runs through a more extensive series of tests, covering "required", "generally necessary", and "optional" system aspects needed by K3s.

Some failures, especially in "optional" aspects, may not actually prevent the cluster from working normally, in the limited ways the LOCKSS system uses Kubernetes. Some of the error messages you might encounter are documented below, but you may need to refer to the official [K3s documentation](#) or use a search engine to look up a specific error message.

### 7.7.1 iptables should be older than v1.8.0, newer than v1.8.3, or in legacy mode

In some instances, you may encounter an error message similar to the following:

```
/usr/sbin iptables v1.8.2 (nf_tables): should be older than v1.8.0, newer than v1.8.3, or in legacy mode (fail)
```

In previous versions of K3s, this error message was also sometimes phrased as `should be older than v1.8.0 or in legacy mode`.

The **install-lockss** script should detect this situation and offer to switch **iptables** to legacy mode via Alternatives (see *Troubleshooting iptables*). If the error above occurs:

- Verify that the *Configuring iptables for K3s* phase of **install-lockss** was not skipped.
- Verify that, if applicable, the proposed **iptables** configuration changes in the *Configuring iptables for K3s* phase of **install-lockss** were not bypassed.
- Using the *Troubleshooting iptables* section as reference, verify that the remediation attempted by **install-lockss** has taken effect.
- Search the [K3s issues database](#) for issues related to **k3s check-config**, **iptables** and your operating system.

### 7.7.2 User namespaces disabled

In some RHEL 7 and CentOS 7 systems, you may receive the following error message:

```
RHEL7/CentOS7: User namespaces disabled; add 'user_namespace.enable=1' to boot commandline (fail)
```

To resolve this issue, see *Enabling User Namespaces in RHEL 7 and CentOS 7*.

---

<sup>2</sup> See *Installing K3s*.

<sup>3</sup> See *Checking the K3s Configuration*.

### 7.7.3 apparmor enabled but apparmor\_parser missing

In some systems where Apparmor is enabled, you may receive the following error:

```
apparmor: enabled, but apparmor_parser missing (fail)
```

To resolve this issue, see *Installing apparmor\_parser*.

### 7.7.4 cgroup hierarchy nonexistent

In some Arch Linux, Debian and Fedora systems, you may see the following error message:

```
cgroup hierarchy: nonexistent?? (fail)
```

K3s supports cgroup2 but **k3s check-config** version 1.21.5+k3s1 (used in LOCKSS 2.0-alpha5) does not process this condition correctly. **This warning can be ignored.**

### 7.7.5 links: aux/iptables should link to iptables-detect.sh

In some Fedora and OpenSUSE systems, you may encounter six related error messages like the following:

```
links: aux/ip6tables should link to iptables-detect.sh (fail)
links: aux/ip6tables-restore should link to iptables-detect.sh (fail)
links: aux/ip6tables-save should link to iptables-detect.sh (fail)
links: aux/iptables should link to iptables-detect.sh (fail)
links: aux/iptables-restore should link to iptables-detect.sh (fail)
links: aux/iptables-save should link to iptables-detect.sh (fail)
```

This is due to a bug in **k3s check-config**<sup>6</sup>, triggered in environments where there is no **iptables** system package installed. **This warning can be ignored.**

### 7.7.6 swap should be disabled

**This warning can be ignored:**

```
swap: should be disabled
```

### 7.7.7 CONFIG\_INET\_XFRM\_MODE\_TRANSPORT missing

**This warning can be ignored:**

```
CONFIG_INET_XFRM_MODE_TRANSPORT: missing
```

---

<sup>6</sup> Reference:

- <https://github.com/k3s-io/k3s/issues/4066>
  - <https://github.com/k3s-io/k3s/issues/4066#issuecomment-925137706>

## 7.8 Troubleshooting OverlayFS with XFS

This section provides troubleshooting information for the `check_xfs` phase of `/installing/installer`.

### 7.8.1 Filesystem backing `/var/lib/rancher` is an XFS filesystem with `ftype=0`

The OverlayFS driver used by K3s will fail and prevent containers from starting, if the filesystem backing `/var/lib/rancher` is an XFS filesystem configured with the legacy `ftype=0` setting that some older Linux systems defaulted to when creating XFS filesystems. To view or verify the setting of `ftype` from an XFS filesystem, use `xfs_info` (see step 1 below).

The recommended way fix to this issue is to reformat the XFS filesystem with `ftype=1`, or with another filesystem such as `ext4`. If this is not possible (e.g., because you are performing an in-place migration from LOCKSS 1.x to 2.x and the filesystem backing `/var/lib/rancher` is shared with other parts of the system), the workaround is to create a new XFS filesystem on another partition or a loopback device backed by a file on an existing filesystem.

Setup of the latter is described below. Commands should be run as the root user<sup>1</sup>.

1. (Optional) Verify the existing XFS filesystem has `ftype=0`:

```
xfs_info BLOCK_DEVICE
```

...where `BLOCK_DEVICE` is the block device of the filesystem backing `/var/lib/rancher`.

**Tip:** Use `findmnt` to determine the block device of the filesystem backing `/var/lib/rancher`:

```
findmnt --target /var/lib/rancher
```

The output should look similar to the following (some parameters may be different):

| TARGET | SOURCE           | FSTYPE | OPTIONS                                                       |
|--------|------------------|--------|---------------------------------------------------------------|
| /      | /dev/mapper/root | xfs    | rw,relatime,attr2,inode64,logbufs=8,logbsize=32k,<br>↪noquota |

In the example above, the block device is `/dev/mapper/root` (i.e., the `SOURCE` column).

The output from `xfs_info` should look similar to the following (some parameters may be different):

```
meta-data=/dev/mapper/root isize=512    agcount=4, agsize=4194304 blks
          =                       sectsz=4096  attr=2, projid32bit=1
          =                       crc=1        finobt=1, sparse=1, rmapbt=0
          =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
data      =                       bsize=4096  blocks=16777216, imaxpct=25
          =                       sunit=0      swidth=0 blks
naming    =version 2               bsize=4096  ascii-ci=0, ftype=0
log       =internal log            bsize=4096  blocks=16384, version=2
          =                       sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none                    extsz=4096  blocks=0, rtextents=0
```

**Note:** In the example above, `ftype=0`. If your output contains `ftype=1`, there is nothing wrong with your XFS filesystem and you should not proceed with the instructions on this page!

<sup>1</sup> See *Running Commands as root*.

2. Stop the K3s service:

```
systemctl stop k3s.service
```

3. Rename the existing `/var/lib/rancher` directory:

```
mv /var/lib/rancher /var/lib/rancher.backup
```

4. Allocate a file for the loopback device: This file will contain the XFS filesystem that will hold the K3s runtime environment and container images. We recommend allocating a minimum of 16GB. Allocate the file on a filesystem with sufficient space.

**Note:** In this and the following examples, we allocate and use the file located at `/var/lib/rancherfs.img`.

```
fallocate -l 16g /var/lib/rancherfs.img
```

5. Create a new XFS filesystem with `ftype` explicitly set to 1:

```
mkfs.xfs -n ftype=1 /var/lib/rancherfs.img
```

You may verify `ftype=1` by running:

```
xfs_info /var/lib/rancherfs.img
```

6. Mount the loopback to `/var/lib/rancher`:

```
mount --mkdir -o loop /var/lib/rancherfs.img /var/lib/rancher
```

7. Edit `/etc/fstab` to make the loopback persistent across reboots: Use your favorite text editor to open `/etc/fstab` and append the following line to it:

```
/var/lib/rancherfs.img /var/lib/rancher xfs loop 0 0
```

8. Copy the existing K3s runtime environment to the new XFS filesystem:

```
cp -r /var/lib/rancher.backup/* /var/lib/rancher/.
```

9. Finally, restart the K3s service:

```
systemctl start k3s.service
```

10. (Optional) If setup of the loopback was successful and K3s is running correctly, you may reclaim space by removing `/var/lib/rancher.backup`:

```
rm -rf /var/lib/rancher.backup
```



## SYSTEM ADMINISTRATION TASKS

This chapter covers some common system administration tasks related to running a LOCKSS system.

Sections include how to run commands as the **root** user, as a privileged user who can become root via **sudo**, or as the **lockss** user; how to update a Linux operating system; and how to install various system utilities like Curl, Wget or HTTPie.

### 8.1 Running Commands as root

Some commands or scripts in this manual are intended to be run as **root**. This section describes two methods for doing so.

#### 8.1.1 Running Commands as root With **sudo**

If you are logged in as a user who can run commands as **root** via **sudo**, simply add the following in front of the command listed in the manual<sup>1</sup>:

```
sudo ...
```

For example, if the command listed in the manual is `iptables -F`, you would type `sudo iptables -F`.

#### 8.1.2 Running Commands Directly as root

If you are logged in as **root** directly, you can simply run the command as listed in the manual, for example `iptables -F`.

---

### 8.2 Running Commands as a Privileged User

Some commands or scripts in this manual are intended to be run as a **privileged user** who can become root and **lockss** via **sudo**.

Compared to running an entire command or script directly as **root**, this approach has the security advantage of being granular -- only those portions of the command or script that require **root** privileges will have **root** privileges, and only those commands that need to read or write files as the **lockss** user will be run as the **lockss** user.

To run a command in this context, simply type the command listed in the manual. Depending on your system's **sudo** configuration, you may be prompted for the user's **sudo** password.

---

<sup>1</sup> Depending on your system's **sudo** configuration, you may be prompted for the user's **sudo** password.

## 8.3 Running Commands as the lockss User

Unless otherwise noted, most commands in this manual are intended to be run as the **lockss** user (oftentimes in the **lockss** user's **lockss-installer** directory). This section describes two methods for doing so.

### 8.3.1 Running Commands as lockss With **sudo**

If you are logged in as a user who can run commands as **lockss** via **sudo**:

- You can start a Bash shell session as the **lockss** user and run any number of commands in it:

1. Run this command<sup>1</sup>:

```
sudo -i -u lockss
```

---

**Tip:** You can also use the slightly shorter version `sudo -iu lockss`.

---

2. Run commands as they are listed in the manual, for example `scripts/start-lockss --wait`.
  3. When you are done, exit the **lockss** shell session by typing `exit` or `logout` or hitting `Ctrl + D`.
- Alternatively, you can use **sudo** to run a single command as the **lockss** user.

Add the following in front of the command listed in the manual<sup>Page 60, 1</sup>:

```
sudo -u lockss ...
```

For example, if the command listed in the manual is `scripts/start-lockss --wait`, you would type `sudo -u lockss scripts/start-lockss --wait`.

### 8.3.2 Running Commands as lockss With **su**

If you are logged in as **root** but your system does not have **sudo** (or does not let **root** use **sudo**), you can use **su** instead:

- You can use **su** to start a Bash shell session as the **lockss** user and run any number of commands in it:

1. Type this command:

```
su lockss
```

2. Run commands as they are listed in the manual, for example `scripts/start-lockss --wait`.
  3. When you are done, exit the **lockss** shell session by typing `exit` or `logout` or hitting `Ctrl + D`.
- Alternatively, you can use **su** to run a single command as the **lockss** user:

Put the command listed in the manual in quotation marks in the following way:

```
su -c '...' lockss
```

For example, if the command to be run as the **lockss** user is `scripts/start-lockss --wait`, you would type `su -c 'scripts/start-lockss --wait' lockss`.

---

<sup>1</sup> Depending on your system's **sudo** configuration, you may be prompted for the user's **sudo** password.



**Caution:** You will need to take care if the command itself contains quotation marks<sup>2</sup>.

## 8.4 Operating System Updates

You will want to update your Linux operating system at different times:

- Before *Upgrading the LOCKSS System*.
- Before *Installing the LOCKSS System* for the first time.
- When security-related updates are released for the Linux kernel (which requires rebooting the machine) or other installed software packages.

Your system may be set up for automatic updates or your system administrator may have policies for which packages can be updated and when.

If you wish to update software packages manually, select your operating system below and follow the instructions as root<sup>1</sup>:

### AlmaLinux OS

To manually update software packages, run this **dnf** command as root:

```
dnf update
```

### Arch Linux

To manually update software packages, run this **pacman** command as root:

```
pacman --sync --refresh --sysupgrade
```

or equivalently:

```
pacman -Syu
```

### CentOS

#### CentOS Stream

To manually update software packages, run this **dnf** command as root:

```
dnf update
```

<sup>2</sup> If the command contains quotation marks, use `-c "..."` instead of `-c '...'`, and add a backslash in front of each double quotation mark in the command (`\` instead of `"`); single quotation marks in the command are unchanged.

<sup>1</sup> See *Running Commands as root*.

### CentOS 7

To manually update software packages, run this **yum** command as root:

```
yum update
```

### Debian

To manually update software packages, run these **apt** commands as root:

```
apt update  
apt upgrade
```

### EuroLinux

#### EuroLinux 8

To manually update software packages, run this **dnf** command as root:

```
dnf update
```

#### EuroLinux 7

To manually update software packages, run this **yum** command as root:

```
yum update
```

### Fedora Linux

To manually update software packages, run this **dnf** command as root:

```
dnf update
```

### Linux Mint

To manually update software packages, run these **apt** commands as root:

```
apt update  
apt upgrade
```

## OpenSUSE

### OpenSUSE Tumbleweed

To manually update software packages, run these **zypper** commands as **root**:

```
zypper refresh  
zypper update
```

### OpenSUSE Leap

To manually update software packages, run these **zypper** commands as **root**:

```
zypper refresh  
zypper update
```

## Oracle Linux

### Oracle Linux 8-9

To manually update software packages, run this **dnf** command as **root**:

```
dnf update
```

### Oracle Linux 7

To manually update software packages, run this **yum** command as **root**:

```
yum update
```

## RHEL

### RHEL 8-9

To manually update software packages, run this **dnf** command as **root**:

```
dnf update
```

### RHEL 7

To manually update software packages, run this **yum** command as root:

```
yum update
```

### Rocky Linux

To manually update software packages, run this **dnf** command as root:

```
dnf update
```

### Scientific Linux

To manually update software packages, run this **yum** command as root:

```
yum update
```

### Ubuntu

To manually update software packages, run these **apt** commands as root:

```
apt update  
apt upgrade
```

---

## 8.5 Installing Curl

*Downloading the LOCKSS Installer* requires [Curl](#), [Wget](#) or [HTTPIe](#). Most typical Linux systems have at least one installed by default. You can check by typing `curl --version`, `wget --version` or `http --version`, and seeing which ones do not output an error message.

This section describes how to install [Curl](#) if necessary.

Select your operating system below and follow the instructions as root<sup>1</sup>:

### AlmaLinux OS

To install Curl with Dnf, run this command (as root):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```

---

<sup>1</sup> See *Running Commands as root*.

## Arch Linux

To install Curl with Pacman, run this command (as **root**):

```
pacman --sync --noconfirm curl
```

or equivalently:

```
pacman -S --noconfirm curl
```

## CentOS

### CentOS Stream

To install Curl with Dnf, run this command (as **root**):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```

## CentOS 7

To install Curl with Yum, run this command (as **root**):

```
yum --assumeyes install curl
```

or equivalently:

```
yum -y install curl
```

## Debian

To install Curl with Apt, follow these steps (as **root**):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes curl
```

or equivalently:

```
apt -y install curl
```

### EuroLinux

#### EuroLinux 8-9

To install Curl with Dnf, run this command (as **root**):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```

#### EuroLinux 7

To install Curl with Yum, run this command (as **root**):

```
yum --assumeyes install curl
```

or equivalently:

```
yum -y install curl
```

### Fedora Linux

To install Curl with Dnf, run this command (as **root**):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```

### Linux Mint

To install Curl with Apt, follow these steps (as **root**):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes curl
```

or equivalently:

```
apt -y install curl
```

## OpenSUSE

### OpenSUSE Tumbleweed

To install Curl with Zypper, follow these steps (as **root**):

1. Run this **zypper** command:

```
zypper refresh
```

2. Run this **zypper** command:

```
zypper --non-interactive install curl
```

or equivalently:

```
zypper -n install curl
```

### OpenSUSE Leap

To install Curl with Zypper, follow these steps (as **root**):

1. Run this **zypper** command:

```
zypper refresh
```

2. Run this **zypper** command:

```
zypper --non-interactive install curl
```

or equivalently:

```
zypper -n install curl
```

## Oracle Linux

### Oracle Linux 8-9

To install Curl with Dnf, run this command (as **root**):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```

### Oracle Linux 7

To install Curl with Yum, run this command (as root):

```
yum --assumeyes install curl
```

or equivalently:

```
yum -y install curl
```

### RHEL

#### RHEL 8-9

To install Curl with Dnf, run this command (as root):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```

### RHEL 7

To install Curl with Yum, run this command (as root):

```
yum --assumeyes install curl
```

or equivalently:

```
yum -y install curl
```

### Rocky Linux

To install Curl with Dnf, run this command (as root):

```
dnf --assumeyes install curl
```

or equivalently:

```
dnf -y install curl
```



## Scientific Linux

To install Curl with Yum, run this command (as root):

```
yum --assumeyes install curl
```

or equivalently:

```
yum -y install curl
```

## Ubuntu

To install Curl with Apt, follow these steps (as root):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes curl
```

or equivalently:

```
apt -y install curl
```

---

## 8.6 Installing HTTPie

*Downloading the LOCKSS Installer* requires [Curl](#), [Wget](#) or [HTTPie](#). Most typical Linux systems have at least one installed by default. You can check by typing `curl --version`, `wget --version` or `http --version`, and seeing which ones do not output an error message.

This section describes how to install [HTTPie](#) if necessary.

Select your operating system below and follow the instructions as root<sup>1</sup>:

### AlmaLinux OS

To install HTTPie with Dnf, run this command (as root):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

---

<sup>1</sup> See *Running Commands as root*.

### Arch Linux

To install HTTPie with Pacman, run this command (as **root**):

```
pacman --sync --noconfirm httpie
```

or equivalently:

```
pacman -S --noconfirm httpie
```

### CentOS

#### CentOS Stream

To install HTTPie with Dnf, run this command (as **root**):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

### CentOS 7

To install HTTPie with Yum, run this command (as **root**):

```
yum --assumeyes install httpie
```

or equivalently:

```
yum -y install httpie
```

### Debian

To install HTTPie with Apt, follow these instructions (as **root**):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes httpie
```

or equivalently:

```
apt -y install httpie
```

## EuroLinux

### EuroLinux 8-9

To install HTTPie with Dnf, run this command (as **root**):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

### EuroLinux 7

To install HTTPie with Yum, run this command (as **root**):

```
yum --assumeyes install httpie
```

or equivalently:

```
yum -y install httpie
```

## Fedora Linux

To install HTTPie with Dnf, run this command (as **root**):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

## Linux Mint

To install HTTPie with Apt, follow these instructions (as **root**):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes httpie
```

or equivalently:

```
apt -y install httpie
```

### OpenSUSE

#### OpenSUSE Tumbleweed

To install HTTPie with Zypper, run these commands (as **root**):

1. Run this **zypper** command:

```
zypper refresh
```

2. Run this **zypper** command:

```
zypper --non-interactive install httpie
```

or equivalently:

```
zypper -n install httpie
```

#### OpenSUSE Leap

To install HTTPie with Zypper, run these commands (as **root**):

1. Run this **zypper** command:

```
zypper refresh
```

2. Run this **zypper** command:

```
zypper --non-interactive install httpie
```

or equivalently:

```
zypper -n install httpie
```

### Oracle Linux

#### Oracle Linux 8-9

To install HTTPie with Dnf, run this command (as **root**):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

## Oracle Linux 7

To install HTTPie with Yum, run this command (as root):

```
yum --assumeyes install httpie
```

or equivalently:

```
yum -y install httpie
```

## RHEL

### RHEL 8-9

To install HTTPie with Dnf, run this command (as root):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

### RHEL 7

To install HTTPie with Yum, run this command (as root):

```
yum --assumeyes install httpie
```

or equivalently:

```
yum -y install httpie
```

## Rocky Linux

To install HTTPie with Dnf, run this command (as root):

```
dnf --assumeyes install httpie
```

or equivalently:

```
dnf -y install httpie
```

### Scientific Linux

To install HTTPie with Yum, run this command (as root):

```
yum --assumeyes install httpie
```

or equivalently:

```
yum -y install httpie
```

### Ubuntu

To install HTTPie with Apt, follow these instructions (as root):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes httpie
```

or equivalently:

```
apt -y install httpie
```

---

## 8.7 Installing Wget

*Downloading the LOCKSS Installer* requires [Curl](#), [Wget](#) or [HTTPie](#). Most typical Linux systems have at least one installed by default. You can check by typing `curl --version`, `wget --version` or `http --version`, and seeing which ones do not output an error message.

This section describes how to install [Wget](#) if necessary.

Select your operating system below and follow the instructions as root<sup>1</sup>:

### AlmaLinux OS

To install Wget with Dnf, run this command (as root):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

---

<sup>1</sup> See *Running Commands as root*.

## Arch Linux

To install Wget with Pacman, run this command (as **root**):

```
pacman --sync --noconfirm wget
```

or equivalently:

```
pacman -S --noconfirm wget
```

## CentOS

### CentOS Stream

To install Wget with Dnf, run this command (as **root**):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

## CentOS 7

To install Wget with Yum, run this command (as **root**):

```
yum --assumeyes install wget
```

or equivalently:

```
yum -y install wget
```

## Debian

To install Wget with Apt, follow these steps (as **root**):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes wget
```

or equivalently:

```
apt -y install wget
```

### EuroLinux

#### EuroLinux 8

To install Wget with Dnf, run this command (as **root**):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

#### EuroLinux 7

To install Wget with Yum, run this command (as **root**):

```
yum --assumeyes install wget
```

or equivalently:

```
yum -y install wget
```

### Fedora Linux

To install Wget with Dnf, run this command (as **root**):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

### Linux Mint

To install Wget with Apt, follow these steps (as **root**):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes wget
```

or equivalently:

```
apt -y install wget
```



## OpenSUSE

### OpenSUSE Tumbleweed

To install Wget with Zypper, run these commands (as root):

1. Run this **zypper** command:

```
zypper refresh
```

2. Run this **zypper** command:

```
zypper --non-interactive install wget
```

or equivalently:

```
zypper -n install wget
```

### OpenSUSE Leap

To install Wget with Zypper, run these commands (as root):

1. Run this **zypper** command:

```
zypper refresh
```

2. Run this **zypper** command:

```
zypper --non-interactive install wget
```

or equivalently:

```
zypper -n install wget
```

## Oracle Linux

### Oracle Linux 8-9

To install Wget with Dnf, run this command (as root):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

### Oracle Linux 7

To install Wget with Yum, run this command (as root):

```
yum --assumeyes install wget
```

or equivalently:

```
yum -y install wget
```

### RHEL

#### RHEL 8-9

To install Wget with Dnf, run this command (as root):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

### RHEL 7

To install Wget with Yum, run this command (as root):

```
yum --assumeyes install wget
```

or equivalently:

```
yum -y install wget
```

### Rocky Linux

To install Wget with Dnf, run this command (as root):

```
dnf --assumeyes install wget
```

or equivalently:

```
dnf -y install wget
```

## Scientific Linux

To install Wget with Yum, run this command (as root):

```
yum --assumeyes install wget
```

or equivalently:

```
yum -y install wget
```

## Ubuntu

To install Wget with Apt, follow these steps (as root):

1. Run this **apt** command:

```
apt update
```

2. Run this **apt** command:

```
apt install --assume-yes wget
```

or equivalently:

```
apt -y install wget
```

---

## 8.8 Resetting the System to a Blank State

**During and after alpha and beta testing**, you may have cause to delete all of the stored content in order to start over. This section provides guidance on how to do this without reinstalling completely from scratch.

LOCKSS 2.x stores content and metadata in several places which depend on how it was configured. We've provided a script to facilitate deleting all the right directories, but to provide flexibility and make it difficult to do this accidentally, the script generates a second script that actually performs the deletions.

The procedure consists of deleting the PostgreSQL data directory (`lockss-stack-postgres-data`) and Solr data directory (`lockss-stack-solr-data`) from the primary content data storage area, and the repository data directory (`lockss-stack-repo-data`) from each content data storage area, all of which must be done after shutting down the system.

### Directions

1. Stop the stack:

```
scripts/stop-lockss
```

2. Generate the deletion script:

```
scripts/upgrades/generate-content-reset-script > /tmp/delcontent
```

3. Examine `/tmp/delcontent` if you wish. Note that there is one **sudo** command because the PostgreSQL files are not owned by the lockss user, which may need modifying in some environments.

4. Make the script executable:

```
chmod +x /tmp/delcontent
```

5. Delete the content:

```
/tmp/delcontent
```

6. We recommend deleting the script after use:

```
rm /tmp/delcontent
```

7. Restart the system:

```
scripts/start-lockss
```

## APPENDIX

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

Sections include security advisories; reference information such as list of ports or component versions; and advanced topics like running the LCAP polling and repair protocol over SSL, working with PostgreSQL, or running the LOCKSS Installer from Git.

### 9.1 Security Advisories

There are no security advisories currently published for LOCKSS 2.0-alpha7.

See also the LOCKSS Documentation Portal's [Security](#) pages.

#### 9.1.1 CVE-2022-39135

*First published: 2023-09-08*

---

**Tip:** LOCKSS 2.x to date (2.0.72-alpha7 as of this writing), when used with its embedded Apache Solr container, is not affected by CVE-2022-39135.

See [CVE-2022-39135](#) in the LOCKSS Documentation Portal's [Security](#) pages.

---

### 9.2 Release Notes

#### 9.2.1 LOCKSS 2.0.72-alpha7

Released: 2023-09-02

LOCKSS 2.0.72-alpha7 is a bug fix release of the LOCKSS 2.0-alpha7 system. If you are running 2.0.71-alpha7, you will need to follow the instructions in *Upgrading the LOCKSS System*.

### Release Notes

- **Bug Fixes**
  - Fixed an errant Java 11 dependency impacting the repository service's reindexing code, typically involved only in the 2.0-alpha5 to 2.0-alpha7 upgrade procedure.

### Component Versions

LOCKSS 2.0.72-alpha7 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.72-alpha7
- LOCKSS Repository Service version 2.14.1
- LOCKSS Configuration Service version 2.8.0
- LOCKSS Poller Service version 2.6.0
- LOCKSS Crawler Service version 1.0.0
- LOCKSS Metadata Extraction Service version 2.7.0
- LOCKSS Metadata Service version 2.6.0
- LOCKSS SOAP Compatibility Service version 1.4.0
- PostgreSQL version 14.7
- Apache Solr version 8.11.2 (custom version 8.11.2-slim-1)
- Pywb version 2.4.2 (custom version 2.4.2-3)
- OpenWayback version 2.4.0 (custom version 2.4.0-5)

### 9.2.2 LOCKSS 2.0.71-alpha7

Released: 2023-08-29

LOCKSS 2.0.71-alpha7 is the first release of the LOCKSS 2.0-alpha7 system.

### Release Notes

- **Features**
  - The major new feature in this release is the LOCKSS Crawler Service, which provides a REST interface to crawling services. This will allow users to integrate external crawlers by defining a Pluggable Crawler Plugin (similar to the way publisher plugins can be defined). Initially, Wget is supported (along with the classic LOCKSS crawler).
  - The provided PostgreSQL container is upgraded to version 14.7. The upgrade script converts existing databases by dumping and reloading them. This may take some time for large databases. **It is very important to run the upgrade script.**
  - NamedPlugin can now be used for both direct deposit and with pluggable crawlers.
  - Added StartupStatus enum to ApiStatus REST response, allows clients to determine when AUs are fully started.
  - Self-generated SSL certificates for admin UI or ServeContent now have 2048-bit keys.
  - Added Flush Artifact Caches button to DebugPanel to assist with benchmarking repository performance.

- AuQuery Web Services response now includes AU access URLs.
- The remaining (deprecated) REST endpoints in service of Web Services queries have been moved to a `/ws/...` path.
- The AUID is displayed in the AU detail page, and available in metadata index status tables.
- Dependency upgrades: Jsoup 1.16.1, Apache Log4J 2.20.0, Apache Commons FileUpload 1.5, Apache Commons Compress 1.23.0, Apache Commons CSV 1.10.0, Apache Commons Codec 1.16.0, Apache Commons IO 2.13.0, Jonix 2023-05, Json-Path 2.8.0, MARC4J 2.9.5, JDBC PostgreSQL client 42.5.0.

- **Bug Fixes**

- Improved robustness handling corrupted WARC files (generally caused by abrupt shutdown).
- Fixed bug causing most services to periodically reload config files even when not changed.
- Reduced spurious undeleted temp files.
- Fixed a unicode normalization vulnerability that might have allowed specially crafted bibliographic info in the title DB to cause the UI to misbehave.
- Fixed config file precedence problem due to inconsistent/wrong loading order.
- Fixed Repository Service startup problem when Solr is not quite ready.
- Fixed a bug preventing hash estimate padding from being fully configurable.
- Error logs sometimes omitted Timestamp messages.
- PostgreSQL logs no longer accumulate forever.

- **LOCKSS Installer changes**

- Now allows the content, state data (including databases), logs, and temporary storage areas to be placed on different devices. We strongly discourage placing the state data and temporary storage areas on network-attached storage such as NFS.
- Allows more end-user control over runtime environment and JVM command line (e.g. to change heap size or add profiling or debugging agents).
- When reconfiguring the system after an upgrade, it is no longer necessary to see and accept the default for each answer. If **configure-lockss** is invoked in replay mode with the `-r` option, all prompts that already have an answer will simply be echoed and the script will proceed to the next prompt. It will be necessary to enter info (or accept the default) only for new prompts added since the previous release.

- **LOCKSS Repository Service changes**

- New endpoints to retrieve artifact data with a streaming response. (The multipart response, still supported, generally precluded clients streaming data into an application, negatively impacting performance.)
- Added `excludeStatusPattern` to `addArtifacts()`, allows skipped artifacts based on HTTP response code in WARC record.
- Added duplicate detection to `addArtifacts()`.
- Removed `isCompressed` argument from `addArtifacts`. The repository service now detects whether the archive file is compressed.
- Performance improvements in repository to spend less time recalculating AU sizes.
- `getStoreageInfo` API now only optionally queries Solr index space, as it is quite slow.

- **Plugin Packager changes**

- Plugin key `plugin_aux_packages` allows declaration of packages that should be included in the plugin JAR.

- Improved plugin validation.
- Validation is now also performed after packaging.

- **Performance**

- Substantial reductions in memory requirements.
- Eliminated many duplicate strings and other objects.
- Periodically clear PDFBox's (monotonically growing) caches.
- Improved hashing performance.
- Removed a slow Solr call that was causing some UI pages to be very slow.
- Reduced extraneous inter-service network traffic. (Unnecessary config file reloading and redundant storing of some state objects.)
- Greatly reduced startup time of services that don't need to load the Title DB.
- Implemented database connection pooling.
- Improved REST connection pooling.
- Reduced start script time.
- Applied PostgreSQL tuning.

## Component Versions

LOCKSS 2.0.71-alpha7 consists of a configurable set of the following components:

- LOCKSS Installer version 2.0.71-alpha7
- LOCKSS Repository Service version 2.14.0
- LOCKSS Configuration Service version 2.8.0
- LOCKSS Poller Service version 2.6.0
- LOCKSS Crawler Service version 1.0.0
- LOCKSS Metadata Extraction Service version 2.7.0
- LOCKSS Metadata Service version 2.6.0
- LOCKSS SOAP Compatibility Service version 1.4.0
- PostgreSQL version 14.7
- Apache Solr version 8.11.2 (custom version 8.11.2-slim-1)
- Pywb version 2.4.2 (custom version 2.4.2-3)
- OpenWayback version 2.4.0 (custom version 2.4.0-5)



## 9.3 Frequently Asked Questions

This section answers some common questions about the LOCKSS system.

I have an existing classic LOCKSS system (version 1.x). Can I upgrade to LOCKSS 2.0-alpha7?

FIXME (beta is less tentative than alpha)

The LOCKSS 2.0-alpha7 release is a technology preview which we are excited to share with the community for testing purposes. It is not yet possible to convert from a classic LOCKSS system (e.g. version 1.77.6) to a LOCKSS 2.0 system for *production* purposes.

However, version 1.77 of the classic LOCKSS system contains a prototype tool to test the migration of archival units (AUs) from a production 1.77 system to a *test* 2.0-alpha7 system, for *testing* purposes. See <https://github.com/lockss/community/wiki/Migration-Tool>.

To help us advance toward the final LOCKSS 2.0 release, please consider installing and running the LOCKSS 2.0-alpha7 release on a test machine and [providing us with your feedback](#).

**I have a LOCKSS system running 2.0-alpha6. Can I upgrade to LOCKSS 2.0-alpha7?**

Yes. You are welcome to wipe your testing data from LOCKSS 2.0-alpha6 and start from scratch, but there is an [upgrade path](#) from LOCKSS 2.0-alpha6.

**Can I use my own PostgreSQL database? Can I use my own Solr database?**

Yes, you can configure the system to use your institution's Postgres database and/or Solr database -- or you can simply let system run included ones locally.

**Can I replay Web content with my own Pywb instance? Can I replay Web content with my own OpenWayback instance?**

Yes, you can configure your own Pywb instance and/or OpenWayback instance to connect directly to the LOCKSS Repository Service -- or you can let the system run included ones locally, or you can choose not to run any Web replay engine at all.

## 9.4 Software License

The LOCKSS system is made available under the terms of the 3-Clause BSD License, a permissive open-source license.

Copyright (c) 2000-2022, Board of Trustees of Leland Stanford Jr. University

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

(continues on next page)

(continued from previous page)

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

For more information, visit the [Software License](#) page on the LOCKSS Web site.

## 9.5 Network Ports

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

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

All ports are in the 24600-24699 range, except the LCAP (LOCKSS polling and repair) port which retains its historical value of 9729, and the OpenWayback port which is 8080.

| Port  | Component                                      |
|-------|------------------------------------------------|
| 8080  | OpenWayback replay engine                      |
| 9729  | LCAP (LOCKSS polling and repair)               |
| 24600 | <i>reserved</i>                                |
| 24602 | PostgreSQL                                     |
| 24603 | Solr                                           |
| 24606 | ActiveMQ                                       |
| 24610 | LOCKSS Repository Service - REST port          |
| 24619 | <i>reserved</i> (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              |
| 24660 | LOCKSS Crawler Service - REST port             |
| 24661 | LOCKSS Crawler Service - UI port               |
| 24670 | LOCKSS Proxy                                   |
| 24671 | <i>reserved</i>                                |
| 24672 | LOCKSS Audit Proxy                             |
| 24673 | <i>reserved</i>                                |
| 24674 | (UDP) ICP server                               |
| 24675 | LOCKSS SOAP Compatibility Service - SOAP port  |
| 24680 | LOCKSS Content Server (ServeContent)           |
| 24681 | Pywb replay engine                             |
| 24682 | <i>reserved</i> (OpenWayback)                  |

## 9.6 Downloading the LOCKSS Installer using git

**Warning:** The official way to download the LOCKSS Installer is now through the LOCKSS Downloader, rather than cloning the LOCKSS Installer as a Git project as in previous releases.

Follow the instructions in *Installing the LOCKSS System* to use the LOCKSS Downloader.

**Tip:** Follow the instructions in /appendix/git to install **git**, if it is not yet available on your system.

Follow these instructions as the `lockss` user<sup>1</sup>:

1. If you have not previously cloned the LOCKSS Installer, run this command to clone it from GitHub:

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

### Troubleshooting

On early CentOS 7 systems (for example CentOS 7.1), you may receive the error message `fatal: unable to access 'https://github.com/lockss/lockss-installer/': Peer reports incompatible or unsupported protocol version`. This is due to outdated network security libraries. Run the command `yum update -y curl nss nss-util nspr` as root to update them, and retry the **git clone** command.

**Tip:** To avoid a harmless Git warning when updating the LOCKSS Installer from GitHub in the future, run this command within the `lockss-installer` directory created by the **git clone** command:

```
git config --local pull.rebase true
```

2. Checkout the 2.0-alpha5 branch of the LOCKSS Installer by running this command within the `lockss-installer` directory created by the **git clone** command earlier:

```
git checkout 2.0-alpha5
```

## 9.7 Working with PostgreSQL

This section of the appendix documents administrative tasks for the embedded PostgreSQL database configured by the LOCKSS 2.x system.

<sup>1</sup> See *Running Commands as the lockss User*

### 9.7.1 Changing the PostgreSQL Database Password

To change the password of the embedded PostgreSQL database, perform the following steps as the `lockss` user<sup>1</sup> in the `lockss` user's `lockss-installer` directory:

1. Ensure the Kubernetes service definitions reflect the current state of the LOCKSS configuration by running:

```
scripts/assemble-lockss
```

2. Start the PostgreSQL database container by running:

```
k3s kubectl apply -n lockss --filename=config/configs/lockss-stack/svcs/lockss-  
↪postgres-service.yaml
```

3. Run the following command to store the name of the PostgreSQL database container into the variable `postgres_pod`:

```
postgres_pod=$(k3s kubectl get pod -n lockss --selector=io.kompose.service=lockss-  
↪postgres-service --output=jsonpath="{.items[0].metadata.name}")
```

4. Run the following command to store the IP of the PostgreSQL database container into the variable `postgres_ip`:

```
postgres_ip=$(k3s kubectl get pod -n lockss --selector=io.kompose.service=lockss-  
↪postgres-service --output=jsonpath="{.items[0].status.podIP}")
```

5. Execute the following command to alter the LOCKSS database user's password, taking care to replace *newpassword* with your new embedded PostgreSQL database password:

```
echo "ALTER USER \"LOCKSS\" WITH PASSWORD 'newpassword'" | k3s kubectl exec  
↪$postgres_pod -n lockss -i -- psql --username=LOCKSS --dbname=postgres
```

Successful execution of the command results in the output `ALTER ROLE`.

6. To verify that the password change worked, run the following command:

```
k3s kubectl exec $postgres_pod -n lockss -it -- psql --username=LOCKSS --  
↪dbname=postgres --host=$postgres_ip
```

and enter *newpassword* at the *Password for user LOCKSS* prompt. If the password change was successful and you enter *newpassword* correctly, you will see a PostgreSQL prompt similar to:

```
psql (9.6.12)  
Type "help" for help.  
  
postgres=#
```

which you can exit by entering `q` or hitting `Ctrl + D`. If the password change was unsuccessful or you do not enter *newpassword* correctly, you will see output similar to:

```
psql: FATAL: password authentication failed for user "LOCKSS"  
command terminated with exit code 2
```

7. Stop the PostgreSQL database container by running this command:

<sup>1</sup> See *Running Commands as the lockss User*.

```
k3s kubectl -n lockss delete service,deployment lockss-postgres-service &&
k3s kubectl -n lockss wait --for=delete pod $postgres_pod --timeout=60s
```

8. Re-run **configure-lockss** so that you can record the new embedded PostgreSQL database password into the configuration of the LOCKSS stack:

```
scripts/configure-lockss
```

See the *PostgreSQL* and *Embedded PostgreSQL Database* sections of *Configuring the LOCKSS System* for details.

## 9.8 LCAP Over SSL

The section explains how to configure secure communication between LOCKSS boxes in a network.

Some LOCKSS networks, such as the Global LOCKSS Network (GLN), are open, in the sense that anyone may join and set up a LOCKSS box to participate in that network. The LOCKSS polling protocol (LCAP) includes several security measures to prevent rogue players from disrupting the network, but it is also possible to create a closed network, where only authorized nodes are allowed to participate. This document describes the steps needed to set up such a network.

In order to ensure that only authorized nodes may participate, each node is issued a private key, and all nodes are provided the set of corresponding public keys. This allows all inter-node communication to be both encrypted and authenticated, using SSL.

---

**Note:** The Classic LOCKSS system (version 1.x) does not support PKCS12, so if building keystores for a network that includes classic LOCKSS nodes, JCEKS should be selected.

---

### 9.8.1 Generating Keystores

The authority in charge of the private LOCKSS network (PLN) must create and distribute Java keystores to all participants. Each box receives two keystores: one containing its own private key (along with a password file containing the secret password for the private key) and another containing the public certificates for each of the boxes in the network. There are two methods available to create these keystores:

- A *Command Line Tool* run in the LOCKSS development environment.
- An *Interactive Tool* invoked in a running LOCKSS node.

In both cases, the admin creating the keystores must know the complete set of hostnames of boxes in the network. More hosts can be added at any time, but a new public keystore must be created and distributed to each box.

#### Command Line Tool

To use the command line tool:

1. Clone the `lockss-core` and `laaws-dev-scripts` projects from GitHub, in sibling directories.
2. Build `lockss-core`.
3. In the root directory of `lockss-core`, run this command:

```
../laaws-dev-scripts/bin/runclass org.lockss.keystore.EditKeyStores -s pubkeystore.  
↪pkcs12 -o keydir box1.pln.org ... boxN.pln.org
```

This will create, in the directory *keydir*, a public keystore named *pubkeystore.pkcs12*, and a pair of files *boxK.pln.org.pkcs12* and *boxK.pln.org.pass* for each one of the *N* host names *box1.pln.org* through *boxN.pln.org*.

4. To add additional hosts, provide the existing public keystore as the value of the *-s* argument, and list the new hosts. The new public keys will be added to the existing public keystore.

### Interactive Tool

1. Bring up a LOCKSS stack, either in the production environment or *runcluster*. In the UI, select *DebugPanel* → *Generate LCAP Keys*.
2. Enter the hostname of each of the LOCKSS boxes in the *Hostnames* text box, then click the *Generate Keystores* button. A *.tgz* or a *.zip* file will be generated and offered for download. This file will contain the private keystore and password file for each host, as well as the shared public keystore.
3. To add additional hosts, use the *Browse* button to supply the existing public keystore, and enter the new hosts in the *Hostnames* text box. The downloaded file will contain the private keystore and password files for each new host, as well as the updated shared public keystore, which must be installed on all hosts.

## 9.8.2 Installing the Keystores

1. **Securely** transmit to each box its two files and the public keystore. Put them in *~lockss/lockss-installer/config/keys*, and set the owner and group to *lockss:lockss* and the permissions to *600*.
2. Restart the stack and check that it is now using SSL. In the UI, select *Daemon Status* → *Comm Channels*. The page should show *SSL: TLSv1.2, Client Auth*.
3. After a few hours, select *Daemon Status* → *Comm Peer Data* to ensure that each box is successfully originating and accepting connections from all the other boxes.