ScienceLogic

SL1 12.1.2 Oracle Linux 8 Conversion Guide

SL1 version 12.1.2

Table of Contents

Oracle Linux 8 Conversion Guide

Overview

As part of the ScienceLogic 12.1 Golden Gate release, the SL1 platform operating system is being upgraded to Oracle Linux 8 (OL8). All new SL1 12.1.0 or later deployments (from the 12.1.0 ISO file) will install the OL8 operating system and do not require any additional conversion activities.

However, existing SL1 customers *upgrading* from an earlier release line, such as SL1 version 11.3, will need to complete the OL8 operating system conversion. The steps of the conversion process are covered in this guide, along with links to relevant procedures in the SL1 Product Documentation and other documents.

> **CAUTION:** To avoid outages or down time, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL8 before attempting to convert a production system.

Workflow for Converting SL1 to Oracle Linux 8

A typical conversion to Oracle Linux 8 for SL1 users includes the following steps:

1. *Plan for the conversion, back up SL1 (optional), and schedule the conversion*.
2. *Upgrade the Database Server and other SL1 appliances to SL1 version 12.1.2*.
3. *Upgrade MariaDB to 10.4.29 on the Database Server, Administration Portal, and any other non-collector SL1 appliance on Oracle Linux 7*.
4. *Convert the Database Server from Oracle Linux 7 to Oracle Linux 8* (which upgrades the Database Server to MariaDB 10.4.31)
5. *Convert the Data Collectors and Message Collectors to Oracle Linux 8*.
6. *Validate the conversion, set up HA/DR, schedule system maintenance, and other post-conversion tasks*.

These steps are covered in the remaining chapters of this guide, with links to the relevant topics in the *SL1 Product Documentation* as needed.

NOTE: If the MariaDB upgrade fails while trying to pull the **epel** repository from any SL1 appliance, you can disable the repository by editing **/etc/yum.repos.d/epel.repo** and setting **enabled=0** in the file. Then run the MariaDB upgrade again.

TIP: For more information, see the *Frequently Asked Questions (FAQs)*, which include the *SL1 Upgrade Paths* and an *Interoperability Matrix*.

CAUTION: All customers must upgrade to SL1 version 12.1.x or later and convert to OL8 by October 31, 2024, or before upgrading to SL1 version 12.2.0. **If you take no action, all SL1 systems with OL7 will continue to run, but ScienceLogic will not support them, and the systems may not be secure.**

Converting PowerFlow to OL8

SL1 and PowerFlow run on separate platforms, and there is no requirement that both SL1 and PowerFlow are running the same version of Oracle Linux. As a result, after you convert your SL1 system to OL8, all PowerFlow instances still operating on OL7 will continue to function normally. You can also have a PowerFlow system running OL8 and an SL1 system running a supported version of SL1 that is still on OL7. For the best results, however, ScienceLogic recommends that you convert both PowerFlow and SL1 to OL8 as soon as possible.

To convert PowerFlow to OL8, you will need to run an upgrade script from ScienceLogic that is available with the ISO file for the PowerFlow version 3.0.0 release. For more information, see *Converting PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.

Converting Restorepoint to OL8

Depending on the Restorepoint version, the application currently runs on either CentOS or OL8. There are no dependencies between Restorepoint and SL1. For more information about converting Restorepoint to OL8, see *Converting Restorepoint to Oracle Linux 8* in the Restorepoint documentation.

Chapter

1

Step 1: Before You Convert to OL8

Overview

This chapter provides an overview of the planning process for converting from Oracle Linux 7 (OL7) to Oracle Linux 8 (OL8). Complete the steps in this section to prepare your SL1 system for the OL8 conversion.

> **CAUTION:** To avoid outages or down time, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL8 before attempting to convert a production system.

This chapter covers the following topics:

Review SL1 Specifications and OL8 System Requirements

SL1 Architecture Specifications by Node

To help you determine the capacity sizing requirements for agents and agentless devices, download the SL1 Architecture Specifications by Node interactive spreadsheet.

System Requirements for Oracle Linux 8

1. Refer to the following page to make sure that your system meets the system requirements for Oracle Linux 8: https://docs.oracle.com/en/operating-systems/oracle-linux/8/install/install-PreparingToInstall.html#prep-install.

2. Refer to the following page to make sure your hardware is supported by Oracle Linux 8: https://linux.oracle.com/hardware-certifications.

> **TIP:** In most cases, having more resources in the system improves the system performance.

For more configuration information, see https://docs.oracle.com/en/operating-systems/oracle-linux/8/install/install-PreparingToInstall.html#syscomp-config.

Considerations for All Conversion Types

Before you start the conversion process, review the following configuration details and take the necessary steps for your SL1 configuration.

- *Considerations for High Availability and Data Recovery (HA/DR)*
- *Space Considerations*
- *PhoneHome Collector Considerations*
- *Considerations for Windows 2008 or Windows 2012*
- *Considerations for SL1 12.1.x and Oracle Linux 8*

Considerations for High Availability and Data Recovery (HA/DR)

The steps in this guide refer to the different SL1 systems as the following:

- Oracle Linux 7: Source and Primary
- Oracle Linux 8: Destination and Secondary

For all conversion types, the Source/Primary system will be either an All-in-One, standalone Database Server, or the current active node in a HA/DR or HA+DR cluster. The Destination/Secondary system will be a freshly installed Oracle Linux 8 system that has been patched to the same version as the Source system. No

configuration, besides patching, if needed, should done to the Destination system until the conversion is completed. This includes setting clustering.

Data conversion from Oracle Linux 7 to Oracle Linux 8 is done using Distributed Replication Block Device (DRBD). If you have an existing High Availability (HA), Disaster Recovery (DR) or a combined HA and DR (HA+DR) cluster, the cluster will be placed into maintenance, and the new Oracle Linux 8 system will be configured to receive replication data. The SL1 platform will not form any automated clustering between Oracle Linux 7 and Oracle Linux 8. During the conversion process, automatic redundancy will be disabled.

Additionally:

- For All-in-One and standalone Database Server systems, you must use a new system to be the Destination.

- For HA, DR, or HA+DR you can use a new system, or you can reuse or reinstall one of the passive nodes in the cluster with Oracle Linux 8. To reuse or reinstall an existing passive node, you must shut down the system before starting *Phase 1* of the procedure.

For all conversion procedures, the Destination system will be a singular system when you complete the conversion:

- An All-in-One Destination system will be a All-in-One system.

- A standalone Database Server, a High Availability (HA), a Disaster Recovery (DR), or a combined HA and DR (HA+DR) system will all be a standalone Database Server.

HA/DR Settings

After you complete the conversion to OL8 and you have verified that the SL1 is working properly, you can implement HA, DR, or HA+DR by following the procedures in the *High Availability & Disaster Recovery Configuration* section of the SL1 Product Documentation.

MariaDB Updates

After the conversion to OL8, all OL8 appliances will be running MariaDB 10.4.31. Until you run the *collector conversion*, your OL7 collectors should be running MariaDB version 10.4.29.

> **NOTE:** If the MariaDB upgrade fails while trying to pull the **epel** repository from any SL1 appliance, you can disable the repository by editing **/etc/yum.repos.d/epel.repo** and setting **enabled=0** in the file. Then run the MariaDB upgrade again.

SL1 Email Settings

To make sure that inbound emails continue to be processed after the conversion to OL8, perform the following steps on the Database Server:

1. In the SL1 user interface, go to the **Email Settings** page (System > Settings > Email).

2. Click **[Save]**.

SL1 Appliances Page

The following image is from an **Appliances** page (System > Settings > Appliances) for an SL1 system where all of the appliances have been converted to OL8:

1.	crucible-g-cdb-172	10.64.164.172	Database	n/a		Database: 10.64.164.172	12.1.2 r465	10.4.31	el8	99,000	n/a	1	Yes	--	No
2.	crucible-giga-ap-183	10.64.164.183	Administration Portal	n/a			12.1.2 r465		el8	n/a	n/a	54	Yes	--	No
3.	crucible-giga-cu-158	10.64.164.158	Data Collection Unit	CUG1			12.1.2 r465	10.4.31	el8	n/a	2941	35	Yes	--	No
4.	crucible-giga-cu-159	10.64.164.159	Data Collection Unit	CUG2			12.1.2 r465	10.4.31	el8	n/a	2941	36	Yes	--	No
5.	crucible-giga-cu-173	10.64.164.173	Data Collection Unit	CUG3			12.1.2 r465	10.4.31	el8	n/a	2941	37	Yes	--	No
6.	crucible-giga-cu-174	10.64.164.174	Data Collection Unit	CUG4			12.1.2 r465	10.4.31	el8	n/a	2942	38	Yes	--	No
7.	crucible-giga-cu-175	10.64.164.175	Data Collection Unit	CUG5			12.1.2 r465	10.4.31	el8	n/a	2941	39	Yes	--	No
8.	crucible-giga-cu-176	10.64.164.176	Data Collection Unit	CUG6			12.1.2 r465	10.4.31	el8	n/a	2941	40	Yes	--	No
9.	crucible-giga-cu-177	10.64.164.177	Data Collection Unit	CUG7			12.1.2 r465	10.4.31	el8	n/a	2941	41	Yes	--	No
10.	crucible-giga-cu-178	10.64.164.178	Data Collection Unit	CUG8			12.1.2 r465	10.4.31	el8	n/a	2941	42	Yes	--	No
11.	crucible-giga-cu-179	10.64.164.179	Data Collection Unit	CUG9			12.1.2 r465	10.4.31	el8	n/a	2941	43	Yes	--	No
12.	crucible-giga-cu-180	10.64.164.180	Data Collection Unit	CUG10			12.1.2 r465	10.4.31	el8	n/a	2941	44	Yes	--	No
13.	crucible-giga-cu-191	10.64.164.191	Data Collection Unit	CUG11			12.1.2 r465	10.4.31	el8	n/a	2941	45	Yes	--	No

Space Considerations

When reinstalling an existing system with Oracle Linux 8 as the migration Destination, keep in mind that partitioning layout and sizes have been altered in OL8. By default, SL1 systems reserve 5% space for emergency allocations. Typically this reservation allows an OL8 system to have the same size physical disk as an OL7 system, which would allow for re-installation onto the same sized disk. In some cases, this reservation might have been used on the OL7 Source system, and the OL8 Destination will not have enough space for the migration to proceed. If this happens, there will be an error presented in Phase 2 of the migration that indicates there is insufficient space to match disk sizes for the migration process.

To check if the space is sufficient:

1. On the OL7 Source system, run the following command to see how much space is allocated to the **/data.local/db** logical volume. Take note of the number.

    ```
    sudo lvdisplay /dev/em7vg/db | grep "Current LE"
    ```

2. On the OL8 Destination system, run the following commands to see how much space is allocated to the **/data.local/db** logical volume and how much free space is in the volume group. Take note of these numbers.

    ```
    sudo lvdisplay /dev/em7vg/db | grep "Current LE"
    ```

    ```
    sudo vgs --noheadings -o vg_free_count
    ```

3. Add the numbers from step 2 and compare them to the number in step 1. If the total from step 2 is greater than the number from step 1, then you have sufficient space. If the total is the same or less, you will need to allocate more space to the OL8 Destination system.

PhoneHome Collector Considerations

When converting SL1 systems that use PhoneHome collectors, it is important to consider how to ensure that collectors can connect to the converted system. The conversion script detects when a system has PhoneHome configured, and the script can convert the PhoneHome configuration so that the converted OL8 system will assume the same role as the Primary system. This preserves the SSH (Secure Shell) keys and the database ID number in the PhoneHome configuration.

Based on your PhoneHome settings, select one of the following options for migrating an SL1 system.

Option 1: Destination server uses current IP address, or secondary HA/DR nodes are behind a NAT with NAT IP redirection

In this configuration, the migration Destination server is a new or existing server that is using the current IP addresses from the secondary HA/DR node, or the secondary HA/DR nodes are behind a NAT and you are redirecting the NAT IP to this migration Destination server.

> **NOTE**: Use this option if you want to migrate an HA/DR stack.

Complete the following steps:

1. When you run the migration script, the script will ask if you want to migrate the PhoneHome information. Answer "no" to this prompt.

2. After the migration is complete, you will need to complete the following steps to manually assign the correct PhoneHome destination to your newly migrated system:

- Determine the ID number of the PhoneHome Destination server that you want to assume by running the following command on the Destination server:

```
silo_mysql -e "SELECT * FROM phonehome.destinations"
```

- Edit the file **/etc/phonehome/config.yaml** by adding an entry like the following:

```
id: <PhoneHome device_id from sql query>
```

> **NOTE:** If you need to create the **config.yaml** file instead of editing an existing version of the file, you will need to adjust the permissions and ownership for the file.

- Allow the PhoneHome port through the firewall with the following commands:

```
firewall-cmd --permanent --add-rich-rule="rule port
port=$<PhoneHome port_number from sql query> protocol=tcp
accept"

firewall-cmd --reload
```

- Start the PhoneHome service:

```
systemctl enable --now phd
```

Option 2: Destination server uses new IP addresses, is not behind a NAT, and is not using a new NAT IP address

In this configuration, the migration Destination server will use new IP addresses, and it is not behind a NAT or is not using a new NAT IP address that is not known to existing systems.

> **NOTE:** Use this option if you want to migrate to OL8 while preserving the existing OL7 system. You should also use this option if you need to move to a different IP space on the network.

For this option, you will need to make sure that your existing SL1 collectors are able to reach the new IP addresses for the migration destination by selecting one of the following options:

- **Option A**: Add a new PhoneHome destination with the new IP address.

  When choosing this option you will need to add the destination and wait for collectors to pick up the data. Then follow the *same steps in option 1* to complete the migration process.

- • **Option B**: Add a new, lower priority secondary PhoneHome destination with the new IP address to an existing PhoneHome destination.

    When choosing this option you will need to add the lower priority secondary PhoneHome destination to the PhoneHome destination that is the migration source. Wait for the destination information to be picked up by the collectors.

    When you run the migration script, the script will ask if you want to migrate the PhoneHome information. Answer "yes" to this prompt.

Option 3: Destination server will replace the OL7 system, and its IP addresses will be updated to match the OL7 system after migration is complete

In this configuration, the migration Destination service will replace the existing OL7 system. You will need to change the IP addresses after the migration so that they match the OL7 system.

> **NOTE:** Use this option if you want to migrate a standalone system from OL7 to OL8.

Complete the following steps:

1. When you run the migration script, the script will ask if you want to migrate the PhoneHome information. Answer "yes" to this prompt.
2. After the migration is complete, shut down or disconnect the OL7 system, and then change the IP addresses for the OL8 system to match the OL7 system.

Option 4: Existing database behind a NAT, will redirect the migration source NAT IP address to the Destination server

In this configuration, your current SL1 Database Server is behind a NAT, and when migration is complete, you want to redirect the migration source NAT IP address to the migration Destination server.

> **NOTE:** Use this option if your migration source and destination are behind a NAT. This option requires you to redirect the migration source NAT IP to the migration destination.

Complete the following steps:

1. When you run the migration script, the script will ask if you want to migrate the PhoneHome information. Answer "yes" to this prompt.
2. After the migration is complete, perform the NAT redirection.

Considerations for Windows 2008 or Windows 2012

If your SL1 system is running Windows 2008 or Windows 2012, and you are using PowerShell collections that have the *Encrypted* field set to *Yes* in the credentials, those collections will stop working. For more information, see *Users with Windows 2008 R2 Servers or Windows 2012 Servers* in the SL1 Product Documentation.

Considerations for SL1 12.1.x and Oracle Linux 8

A known issue might cause high swap usage in excess of 95% to be observed on appliance types running SL1 12.1.x and Oracle Linux 8. This impacts all appliance types, but is most frequently observed on Database Servers or appliances that are under heavy memory pressure. If you encounter this issue after converting to OL8, see: https://support.sciencelogic.com/s/article/11598.

Back Up Current Data in SL1 (Optional)

As a safety precaution, ScienceLogic recommends that you set up an SL1 backup plan for the existing SL1 application (Oracle Linux 7 version). This step is not required, but recommended.

ScienceLogic recommends that you create at a minimum a **configuration** backup of your SL1 system. If the database data must be retained, a **full** backup of SL1 is required. For more information, see *Types of SL1 Backups* in the SL1 Product Documentation.

> **NOTE:** If the environment is considered a "throw-away" configuration, full data backups are not required.

After you determine what type of backup is needed, complete the steps as outlined in the *Backup Management* topic in the SL1 Product Documentation. The backup process requires that you designate an alternative storage location, like local storage or Amazon S3 bucket.

The following links refer to the SL1 Product Documentation topics related to backing up your SL1 data.

To back up your current data in SL1:

1. *Determine the type of SL1 backup you want to use*

2. *Prepare SL1 for the backup*

3. *Create a backup credential*

4. Run the backup that is relevant to your SL1 configuration:

     ◦ *Configuration Backup*

     ◦ *Full Backup*

     ◦ *Disaster Recovery Backup*

Schedule the Conversion

You can schedule the following steps in one or multiple maintenance window periods over the course of about two or three weeks:

1. *Upgrade to SL1 12.1.2* (pause a week or so for testing).
2. *Convert the Database Server to Oracle Linux 8* (pause a week or so for testing). You should also *re-ISO the Administration Portal for your system*.
3. *Convert Data Collectors and Message Collectors to Oracle Linux 8.*

For more information about the potential downtime for the conversion, see *this FAQ*.

The following image gives a high-level overview of the conversion process:

Upgrade SL1 to Version 12.1.2

> **NOTE**: OL7 to OL8 conversions are only supported on the 12.1 release line. Future major releases, including 12.2, which is already available, will *only* support OL8.

The currently supported versions of SL1 running OL7 are versions 11.3 and 12.1 (plus 12.2, which is already on OL8). If you are running a supported version of SL1 that is older than version 12.1.1 or 12.1.2, you will need to upgrade SL1 as part of the standard system update procedure, which is also known as the "silo_update" patch process.

> **IMPORTANT:** You must first *import* version 12.1.0.2 to your SL1 system and then upgrade to 12.1.2 before you can begin the Oracle Linux 8 operating system conversion. You will need a 30-minute pause on 12.1.0.2 so the upgrade procedure can import the RPM file for OL7 and OL8. No 12.1.0.2 staging or deployment is needed.

For detailed information about upgrading from a supported version of SL1, see *Updating SL1* in the SL1 Product Documentation. The SL1 Oracle Linux 8 verification process will validate if the SL1 version has fully completed the update process before proceeding.

For upgrade steps based on supported upgrade paths, including upgrade paths for versions of SL1 that are no longer supported, see the *Important Upgrade Notes for SL1 Golden Gate 12.1.2* section in the 12.1.2 release notes.

> **CAUTION:** Changes to how execution environments deploy in SL1 version 12.1.2 causes some execution environments to no longer work properly post-upgrade, which in turn can cause multiple PowerPacks to not run after you upgrade. For the 12.1.2 release, you should download and install the "SL1: Execution Environment Check" v100 PowerPack and follow the steps in the section *Checking Your SL1 Execution Environments Before You Upgrade* to ensure that your execution environments and PowerPacks will continue to work as intended post-upgrade. This PowerPack can be run on SL1 versions 11.3.1 to 12.1.2, so ScienceLogic recommends performing this check **before you upgrade to 12.1.2**.

After upgrading SL1 to version 12.1.2, go to *The Conversion Process* chapter to continue converting your Database Server and collectors to Oracle Linux 8.

Chapter

2

Step 2: Converting to OL8

Overview

The conversion process for Oracle Linux 8 (OL8) is different based on your SL1 configuration, and the various configuration types are covered in this chapter. If your configuration is not listed here, contact your ScienceLogic Customer Success Manager (CSM). If you encounter issues during the conversion, contact ScienceLogic Support.

For all conversion procedures, the Destination system will be a singular system when you complete the conversion:

- An All-in-One Destination system will be a All-in-One system.
- A standalone Database Server, a High Availability (HA), a Disaster Recovery (DR), or a combined HA and DR (HA+DR) system will all be a standalone Database Server.

HA/DR Settings

After you complete the conversion to OL8 and you have verified that the SL1 is working properly, you can implement HA, DR, or HA+DR by following the procedures in the *High Availability & Disaster Recovery Configuration* section of the SL1 Product Documentation.

MariaDB Updates

After the conversion to OL8, all OL8 appliances will be running MariaDB 10.4.31. Until you run the *collector conversion*, your OL7 collectors should be running MariaDB version 10.4.29.

> **NOTE:** If the MariaDB upgrade fails while trying to pull the **epel** repository from any SL1 appliance, you can disable the repository by editing **/etc/yum.repos.d/epel.repo** and setting **enabled=0** in the file. Then run the MariaDB upgrade again.

SL1 Email Settings

To make sure that inbound emails continue to be processed after the conversion to OL8, perform the following steps on the Database Server:

1. In the SL1 user interface, go to the **Email Settings** page (System > Settings > Email).
2. Click **[Save]**.

SL1 Appliances Page

The following image is from an **Appliances** page (System > Settings > Appliances) for an SL1 system where all of the appliances have been converted to OL8:

1.	crucible-g-cdb-172	10.64.164.172	Database	n/a		Database: 10.64.164.172	12.1.2 r465	10.4.31	el8	99,000	n/a	1	Yes	--	No
2.	crucible-giga-ap-183	10.64.164.183	Administration Portal	n/a			12.1.2 r465		el8	n/a	n/a	54	Yes	--	No
3.	crucible-giga-cu-158	10.64.164.158	Data Collection Unit	CUG1			12.1.2 r465	10.4.31	el8	n/a	2941	35	Yes	--	No
4.	crucible-giga-cu-159	10.64.164.159	Data Collection Unit	CUG2			12.1.2 r465	10.4.31	el8	n/a	2941	36	Yes	--	No
5.	crucible-giga-cu-173	10.64.164.173	Data Collection Unit	CUG3			12.1.2 r465	10.4.31	el8	n/a	2941	37	Yes	--	No
6.	crucible-giga-cu-174	10.64.164.174	Data Collection Unit	CUG4			12.1.2 r465	10.4.31	el8	n/a	2942	38	Yes	--	No
7.	crucible-giga-cu-175	10.64.164.175	Data Collection Unit	CUG5			12.1.2 r465	10.4.31	el8	n/a	2941	39	Yes	--	No
8.	crucible-giga-cu-176	10.64.164.176	Data Collection Unit	CUG6			12.1.2 r465	10.4.31	el8	n/a	2941	40	Yes	--	No
9.	crucible-giga-cu-177	10.64.164.177	Data Collection Unit	CUG7			12.1.2 r465	10.4.31	el8	n/a	2941	41	Yes	--	No
10.	crucible-giga-cu-178	10.64.164.178	Data Collection Unit	CUG8			12.1.2 r465	10.4.31	el8	n/a	2941	42	Yes	--	No
11.	crucible-giga-cu-179	10.64.164.179	Data Collection Unit	CUG9			12.1.2 r465	10.4.31	el8	n/a	2941	43	Yes	--	No
12.	crucible-giga-cu-180	10.64.164.180	Data Collection Unit	CUG10			12.1.2 r465	10.4.31	el8	n/a	2941	44	Yes	--	No
13.	crucible-giga-cu-191	10.64.164.191	Data Collection Unit	CUG11			12.1.2 r465	10.4.31	el8	n/a	2941	45	Yes	--	No

> **CAUTION:** To avoid outages or downtime, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL8 before attempting to convert a production system.

This chapter covers the following configuration types for converting to Oracle Linux 8:

- *On-premises Lab or Test Environment: Single Database Server (All-in-One)*
- *On-premises Production Environment: Multiple Database Servers with HA/DR*
- *Customer-hosted Cloud Platform as a Service (PaaS)*
- *ScienceLogic-hosted Cloud Software as a Service (SaaS)*
- *Extended Architecture (AWS)*
- *Extended Architecture (on-premises)*
- *Message Collectors and Data Collectors*

On-premises Lab or Test Environment: Single Database Server/All-In-One

The SL1 customer on-premises configuration with a single central Database Server (also called a Central Database or CDB) is typically used for feature development, laboratory tests, non-High-Availability (non-HA), and proof-of-concept deployments.

You can deploy this configuration type as part of an SL1 *All-in-One (AIO) system*, or as an SL1 *distributed architecture* system.

> **IMPORTANT:** ScienceLogic recommends that you read through this entire chapter before starting the conversion. Contact your ScienceLogic representative for assistance or more detailed guidance, if needed.

Conversion Steps

> **IMPORTANT:** Before converting to OL8, you will first need to upgrade to SL1 version 12.1.x. For more information, see *Upgrade SL1 to Version 12.1.2*.

The Oracle Linux 8 conversion process for a single Database Server and/or an All-In-One Appliance includes the following steps:

1. *Review the prerequisites*.
2. *Phase 1: Prepare the Primary system for data replication*.
3. *Phase 2: Prepare the Secondary (Destination) system for data replication*.
4. *Phase 3: Wait for data replication to finish*.
5. *Phase 4: Gracefully demote the Primary (Source) system*.
6. *Phase 5: Gracefully promote the Secondary (Destination) system*.
7. *Validation: Ensure the conversion was successful and the system is functioning*.

> **NOTE:** You can also convert a system to OL8 by running the steps below. You might consider this option if you only have one system and do not have a spare system to make the transition, or if your existing system does not have enough resources to set up a new SL1 system that is the same size or larger than the existing SL1 system.
>
> - *Upgrade SL1 to version 12.1.2 or later.*
> - *Create a full backup and a configuration backup of the SL1 system.*

> - *Install the Database Server from the OL8 ISO and then upgrade it to the same version as the previous SL1 version.*
> - *Restore the full backup and configuration backup of your SL1 system.*

Prerequisites for a Single Database Server

- The Primary system should be healthy and meet current requirements for CPU/Memory/Disk.
- The new (Secondary) system has the same or greater specifications as the Primary, installed and patched to the same version as the Primary. No configuration, besides patching (if needed) should done to this new Secondary system.
- Both Primary and Secondary systems must have a network interface that is in the same broadcast domain (subnet) to use for the data transfer.
  - This interface can be the Primary network interface or a Secondary one. If possible, a Secondary network interface connected with a cross-over cable is preferred.
  - The recommended network interface speed is 10 GB, with a minimum speed of 1 GB.

NOTE: ScienceLogic assumes that you will be using the same IP address and name for the Oracle Linux 8 system that you used for the Oracle Linux 7 system. If you change the IP address, you will need to get a new license from ScienceLogic. For more information, see *Licensing and Configuring a Database Server or All-In-One Appliance*.

Additional Considerations

For standalone Database Server systems, you must use a new system for the Destination.

Notes about the conversion process:

- Converting to Oracle Linux 8 requires downtime during upgrade, conversion, and restore.
- The data resync length varies based on the size of the Database Server.

These steps refers to the SL1 systems as the following:

- Oracle Linux 7: Primary
- Oracle Linux 8: Secondary

All-In-One Appliance or Database Server Redeploy

These systems are the most sensitive and critical systems for the SL1 platform. Being able to plan an "upgrade" and meet a time window with a rollback possibility is critical.

- ScienceLogic ensures that the system will be updated with partitioning, which is needed for future growth.

- No configuration data is left over from uninstalled Oracle Linux 7 packages. All previous, outdated, or obsolete code will be removed.

- Data is replicated using the existing DRBD service to the Oracle Linux 8 node.

- Timing for the cutover can be scheduled according to your requirements, so you can perform the Oracle Linux 8 conversion at any time after upgrading SL1. The cutover time to the Oracle Linux 8 system is short.

- If you choose to do so, you can keep the previous system in place, just in case the cutover fails and a rollback is needed.

Phase 1: Prepare the Primary System for Data Replication

After you upgrade SL1 to version 12.1.x, you will need to run the Oracle Linux 8 conversion script, **silo-migrate**. This system administration script will be accessible from the command line.

To run the conversion script:

1. SSH in to the Primary (Oracle Linux 7) system and elevate to root:

    ```
    sudo -s
    ```

2. Run the conversion script:

    ```
    silo-migrate
    ```

3. Review the information from the script and press "y" to confirm that you have a full *backup* and that you understand the risks of conversion:

```
This script will assist in the migration of your SL1 AIO/Database from
the Oracle Linux 7 operating system to the Oracle Linux 8 operating
system. This migration is done using our block level replication tool
(DRBD). This process is conducted in the following phases:

  - Phase 1, prepare source system for data replication
  - Phase 2, prepare destination system for data replication
  - Phase 3, wait for data replication to complete
  - Phase 4, gracefully demote and stop source system
  - Phase 5, gracefully promote and start destination system

Please ensure you are familiar with the migration documentation, and
risks of this procedure, all prerequisites are met, and ensure you have
a valid full backup before proceeding.

I have confirmed I have a valid full backup, and I understand the risks
and am ready to proceed. (y/n)
```

The script performs several pre-checks on the system to determine if there are known issues that will interfere with the conversion process. If any issues are encountered, the script will exit.

4. Review the feedback from Phase 1 of the script and press "y" to confirm that your system was detected correctly (as either an `All-in-One` or `Database` appliance model and `Standalone` redundancy architecture):

```
Begining phase 1 - Prepare this system to as the conversion source


This system has been detected as running on OL7. This will be the
PRIMARY
system that will provide the data to be replicated to a new OL8
system.
Your system has been detected as the following:
  Appliance Model: All-in-One
  Redundancy Architecture: Standalone
Is this correct? (y/n)
```

> **NOTE**: For an All-In-One system, you will be warned that you need to stop the SL1 services and MariaDB to the system for conversion.

5. Enter the number listed next to the IP address on this system that will be used for the DRBD replication traffic.

6. Enter the IP address for the DRBD traffic to the new OL8 system.

7. Confirm that you are ready to have the system services shut down and reconfigured for conversion. Answer **yes** to proceed and answer **no** if you wish to exit. The system stops the SL1 services and MariaDB, and it is reconfigured for conversion to Oracle Linux 8.

8. After the system is reconfigured, it will resume operations. Phase 1 of the conversion is done.

Phase 2: Prepare the Secondary System for Data Replication

Run the conversion script:

1. SSH in to the Secondary (Oracle Linux 8) system and elevate to root:

```
sudo –s
```

2. Run the conversion script:

```
silo-migrate
```

3. Enter an IP address for the Primary (Oracle Linux 7) system that is reachable by this system. This will connect to MariaDB and retrieve the conversion settings.

> **NOTE:** If the username and password in the local **silo.conf** cannot be used to connect to the Primary system, you will be prompted again for the IP address and then a username and password. This username and password are for connecting to MariaDB. After a connection is established, the local **silo.conf** will be updated with the correct information to authenticate automatically.

After it is connected, the script will retrieve the required information about the conversion, and prompt to shut down services on this system to reconfigure it to receive the data.

4. Confirm that you are ready to have the system services shut down and reconfigured for conversion. Type **y** to proceed and answer **n** if you wish to exit. If you type **y**, the system will reconfigure all the data to be migrated with DRBD.

Phase 2 of the conversion is complete, and data will start replicating. The speed of the replication will depend on system utilization and network bandwidth.

> **CAUTION:** Systems should not be rebooted or restarted while the conversion procedure is being executed. If the system is rebooted, it may take manual intervention to resume the synchronization process.

Phase 3: Wait for Data Replication to Finish

While waiting for the data replication process to finish, you can check the status by running the following command on either system:

```
drbdadm status
```

Review the output to see the percentage completed (the `done` value):

```
r0 role:Primary
  disk:UpToDate
  peer role:Secondary
    replication:SyncSource peer-disk:Inconsistent done:21.85
```

When the system is fully replicated and in sync, the script will indicate that the peer-disk is up to date:

```
r0 role:Primary
  disk:UpToDate
  peer role:Secondary
    replication:Established peer-disk:UpToDate
```

Phase 4: Gracefully Demote the Primary (Source) System

Run the conversion script:

1. SSH in to the Primary (Oracle Linux 7) system and elevate to root:

    ```
    sudo –s
    ```

2. Run the conversion script:

    ```
    silo-migrate
    ```

3. The system will be checked for replication status, and you will be prompted to proceed with halting the system. Type **y** to proceed, and the system will gracefully halt. Type **n** if you wish to exit.

4. Wait for the system to halt. When it does, Phase 4 of the conversion is complete.

Phase 5: Gracefully Promote the Secondary (Destination) System

Run the conversion script:

1. SSH in to the Secondary (Oracle Linux 8) system and elevate to root:

    ```
    sudo –s
    ```

2. Run the conversion script:

    ```
    silo-migrate
    ```

3. The system will be checked for replication status, and you will be prompted to proceed with halting the system. Type **y** to proceed, and the system will reconfigure and start all SL1 services. Type **n** if you wish to exit.

4. Wait for the system to restart and be accessible. When it does, Phase 5 of the conversion is complete.

Validation

Ensure the conversion was successful and the SL1 system is functioning by running the tasks in *After You Convert to OL8*.

> **NOTE:** You might experience an issue where the spool service is disconnected after the Database Server conversion to OL8. If this occurs, after the conversion, you can use SSH to access the Database Server and use the following command to restart the spool service:

```
sudo systemctl restart siloupdate-spool.service
```

Not doing so might cause you to experience an error while performing the OL8 conversion for your collector appliances.

Additional Configuration

As needed, see the following topics to convert collectors and other ScienceLogic products to OL8:

- *Convert the Data Collectors and Message Collectors to Oracle Linux 8*. You should also *re-ISO the Administration Portal for your system* if you have not yet done so.

- *Convert PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.

- *Converting Restorepoint to Oracle Linux 8 (OL8)* in the Restorepoint documentation.

On-premises Production Environment: Multiple Database Servers with HA/DR

Data conversion from Oracle Linux 7 to Oracle Linux 8 is done using Distributed Replication Block Device (DRBD). If you have an existing High Availability (HA), Disaster Recovery (DR) or a combined HA and DR (HA+DR) cluster, the cluster will be placed into maintenance, and the new Oracle Linux 8 system will be configured to receive replication data. The SL1 platform will not form any automated clustering between Oracle Linux 7 and Oracle Linux 8. During the conversion process, automatic redundancy will be disabled.

Conversion Steps

> **IMPORTANT:** Before converting to Oracle Linux 8, you will first need to upgrade to SL1 version 12.1.2. For more information, see *Upgrade SL1 to Version 12.1.2*.

The Oracle Linux 8 conversion process for multiple Database Servers includes the following steps:

1. *Review the prerequisites*.
2. *Phase 1: Prepare the Primary (Source) system for data replication*.
3. *Phase 2: Prepare the Secondary (Destination) system for data replication*.
4. *Phase 3: Wait for data replication to finish*.
5. *Phase 4: Gracefully demote the Primary (Source) system*.
6. *Phase 5: Gracefully promote the Secondary (Destination) system*.
7. *Validation: Ensure the conversion was successful and the system is functioning*.

Prerequisites for Multiple Database Servers with HA/DR

> **IMPORTANT:** ScienceLogic recommends that you read through this entire chapter before starting the conversion. Contact your ScienceLogic representative for assistance or more detailed guidance, if needed.

The following settings are required:

- The Primary system should be healthy and meet current requirements for CPU/Memory/Disk.
- **For a High Availability (HA) configuration**, make sure that the HA cluster is healthy, with current CRM templates installed.
- **For a High Availability and Disaster Recovery (HA+DR) configuration**, make sure the HA+DR cluster is healthy, with current CRM templates installed.

- The new (Secondary) system has the same or greater specifications as the Primary, installed and patched to the same version as the Primary. No configuration, besides patching (if needed) should done to this new Secondary system.

- Both Primary and Secondary systems must have a network interface that is in the same broadcast domain (subnet) to use for the data transfer.

  ◦ This interface can be the Primary network interface or a Secondary one. If possible, a Secondary network interface connected with a cross-over cable is preferred.

  ◦ The recommended network interface speed is 10 GB, with a minimum speed of 1 GB.

- **For an HA+DR configuration**, select which node you would like to replace in the HA+DR cluster and shut down the node before you start Phase 1:

  ◦ ScienceLogic recommends that you replace the peer HA node, as this will let you reuse the same IP address for the existing system.

  ◦ Replacing the DR node will require a system that is local to the Primary (Oracle Linux 7) system to meet the network connection requirements.

Notes about the conversion process:

- Converting to Oracle Linux 8 involves a round-robin redeploy process that limits downtime to failover intervals.

- The data resync length varies based on the size of the Database Server.

Considerations for High Availability and Data Recovery (HA/DR)

The steps in this guide refer to the different SL1 systems as the following:

- Oracle Linux 7: Source and Primary

- Oracle Linux 8: Destination and Secondary

For all conversion types, the Source/Primary system will be either an All-in-One, standalone Database Server, or the current active node in a HA/DR or HA+DR cluster. The Destination/Secondary system will be a freshly installed Oracle Linux 8 system that has been patched to the same version as the Source system. No configuration, besides patching, if needed, should done to the Destination system until the conversion is completed. This includes setting clustering.

Data conversion from Oracle Linux 7 to Oracle Linux 8 is done using Distributed Replication Block Device (DRBD). If you have an existing High Availability (HA), Disaster Recovery (DR) or a combined HA and DR (HA+DR) cluster, the cluster will be placed into maintenance, and the new Oracle Linux 8 system will be configured to receive replication data. The SL1 platform will not form any automated clustering between Oracle Linux 7 and Oracle Linux 8. During the conversion process, automatic redundancy will be disabled.

Additionally:

- **For All-in-One and standalone Database Server systems**, you must use a new system to be the Destination.

- **For HA, DR, or HA+DR**, you can use a new system, or you can reuse or reinstall one of the passive nodes in the cluster with Oracle Linux 8. To reuse or reinstall an existing passive node, you must shut down the system before starting *Phase 1* of the procedure.

For all conversion procedures, the Destination system will be a singular system when you complete the conversion:

- An All-in-One Destination system will be a All-in-One system.
- A standalone Database Server, a High Availability (HA), a Disaster Recovery (DR), or a combined HA and DR (HA+DR) system will all be a standalone Database Server.

HA/DR Settings

After you complete the conversion to OL8 and you have verified that the SL1 is working properly, you can implement HA, DR, or HA+DR by following the procedures in the *High Availability & Disaster Recovery Configuration* section of the SL1 Product Documentation.

MariaDB Updates

After the conversion to OL8, all OL8 appliances will be running MariaDB 10.4.31. Until you run the *collector conversion*, your OL7 collectors should be running MariaDB version 10.4.29.

> **NOTE:** If the MariaDB upgrade fails while trying to pull the **epel** repository from any SL1 appliance, you can disable the repository by editing **/etc/yum.repos.d/epel.repo** and setting **enabled=0** in the file. Then run the MariaDB upgrade again.

SL1 Email Settings

To make sure that inbound emails continue to be processed after the conversion to OL8, perform the following steps on the Database Server:

1. In the SL1 user interface, go to the **Email Settings** page (System > Settings > Email).
2. Click **[Save]**.

SL1 Appliances Page

The following image is from an **Appliances** page (System > Settings > Appliances) for an SL1 system where all of the appliances have been converted to OL8:

1.	crucible-g-cdb-172	10.64.164.172	Database	n/a		Database: 10.64.164.172	12.1.2 r465	10.4.31	el8	99,000	n/a	1	Yes	--	No
2.	crucible-giga-ap-183	10.64.164.183	Administration Portal	n/a			12.1.2 r465		el8	n/a	n/a	54	Yes	--	No
3.	crucible-giga-cu-158	10.64.164.158	Data Collection Unit	CUG1			12.1.2 r465	10.4.31	el8	n/a	2941	35	Yes	--	No
4.	crucible-giga-cu-159	10.64.164.159	Data Collection Unit	CUG2			12.1.2 r465	10.4.31	el8	n/a	2941	36	Yes	--	No
5.	crucible-giga-cu-173	10.64.164.173	Data Collection Unit	CUG3			12.1.2 r465	10.4.31	el8	n/a	2941	37	Yes	--	No
6.	crucible-giga-cu-174	10.64.164.174	Data Collection Unit	CUG4			12.1.2 r465	10.4.31	el8	n/a	2942	38	Yes	--	No
7.	crucible-giga-cu-175	10.64.164.175	Data Collection Unit	CUG5			12.1.2 r465	10.4.31	el8	n/a	2941	39	Yes	--	No
8.	crucible-giga-cu-176	10.64.164.176	Data Collection Unit	CUG6			12.1.2 r465	10.4.31	el8	n/a	2941	40	Yes	--	No
9.	crucible-giga-cu-177	10.64.164.177	Data Collection Unit	CUG7			12.1.2 r465	10.4.31	el8	n/a	2941	41	Yes	--	No
10.	crucible-giga-cu-178	10.64.164.178	Data Collection Unit	CUG8			12.1.2 r465	10.4.31	el8	n/a	2941	42	Yes	--	No
11.	crucible-giga-cu-179	10.64.164.179	Data Collection Unit	CUG9			12.1.2 r465	10.4.31	el8	n/a	2941	43	Yes	--	No
12.	crucible-giga-cu-180	10.64.164.180	Data Collection Unit	CUG10			12.1.2 r465	10.4.31	el8	n/a	2941	44	Yes	--	No
13.	crucible-giga-cu-191	10.64.164.191	Data Collection Unit	CUG11			12.1.2 r465	10.4.31	el8	n/a	2941	45	Yes	--	No

Space Considerations

When reinstalling an existing system with Oracle Linux 8 as the migration Destination, keep in mind that partitioning layout and sizes have been altered in OL8. By default, SL1 systems reserve 5% space for emergency allocations. Typically this reservation allows an OL8 system to have the same size physical disk as an OL7 system, which would allow for re-installation onto the same sized disk. In some cases, this reservation might have been used on the OL7 Source system, and the OL8 Destination will not have enough space for the migration to proceed. If this happens, there will be an error presented in Phase 2 of the migration that indicates there is insufficient space to match disk sizes for the migration process.

To check if the space is sufficient:

1. On the OL7 Source system, run the following command to see how much space is allocated to the **/data.local/db** logical volume. Take note of the number.

   ```
   sudo lvdisplay /dev/em7vg/db | grep "Current LE"
   ```

2. On the OL8 Destination system, run the following commands to see how much space is allocated to the **/data.local/db** logical volume and how much free space is in the volume group. Take note of these numbers.

   ```
   sudo lvdisplay /dev/em7vg/db | grep "Current LE"
   ```

   ```
   sudo vgs --noheadings -o vg_free_count
   ```

3. Add the numbers from step 2 and compare them to the number in step 1. If the total from step 2 is greater than the number from step 1, then you have sufficient space. If the total is the same or less, you will need to allocate more space to the OL8 Destination system.

PhoneHome Collector Considerations

When converting systems that use PhoneHome collectors, it is important to consider how to ensure that collectors can connect to the converted system. The conversion script detects when a system has PhoneHome configured, and the script can convert the PhoneHome configuration so that the converted Oracle Linux 8 system will assume the same role as the Primary system. This preserves the SSH (Secure Shell) keys and the database ID number in the PhoneHome configuration.

When converting a system with PhoneHome, be sure to consider how to maintain connectivity to the new system. Consider the following methods to ensure that PhoneHome continues to function:

- Convert the PhoneHome settings when prompted in the script. After the conversion is completed, change the IP of the system or redirect NAT (if you are using NAT) so that the new OL8 database answers the SSH connections of the OL7 Source system.

- Add a second lower-priority destination for the OL7 Source system that has the IP of the Oracle Linux 8 Destination system. When the conversion occurs, the Source system is taken offline, and the Destination system is brought online. The PhoneHome collectors will try the new Destination and connect.

- Manually configure the OL8 system to assume the Destination ID number of another PhoneHome database.

Phase 1: Prepare the Primary System for Data Replication

After you upgrade SL1 to version 12.1.x, you will need to run the Oracle Linux 8 conversion script, **silo-migrate**. This system administration script will be accessible from the command line.

> IMPORTANT: Before you start this process, select which node you would like to replace in the HA+DR cluster and shut down the node. ScienceLogic recommends that you replace the peer HA node, as this will allow reuse of the same IP address of the existing system. Alternately, you can use the same VMs you were using, but you will need to rebuild the VM with the SL1 12.1.x ISO (which is running OL8), using the same IP address. You can also use a new VM built with the SL1 12.1.x ISO and another IP address. In all cases, the **silo-migrate** script requires that you have a destination system with OL8 running on it (either a rebuilt secondary HA node or a new OL8 node).

Run the conversion script:

1. SSH in to the Primary (Oracle Linux 7) system and elevate to root:

   ```
   sudo –s
   ```

2. Run the conversion script:

   ```
   silo-migrate
   ```

3. Review the information from the script and press "y" to confirm that you have a full *backup* and that you understand the risks of conversion:

   ```
   This script will assist in the migration of your SL1 AIO/Database from
   the Oracle Linux 7 operating system to the Oracle Linux 8 operating
   system. This migration is done using our block level replication tool
   (DRBD). This process is conducted in the following phases:

     - Phase 1, prepare source system for data replication
     - Phase 2, prepare destination system for data replication
     - Phase 3, wait for data replication to complete
     - Phase 4, gracefully demote and stop source system
     - Phase 5, gracefully promote and start destination system

   Please ensure you are familiar with the migration documentation, and
   risks of this procedure, all prerequisites are met, and ensure you have
   a valid full backup before proceeding.

   I have confirmed I have a valid full backup, and I understand the risks
   and am ready to proceed. (y/n)
   ```

4. Review the feedback from Phase 1 of the script and press "y" to confirm that your system was detected correctly (as either an `Database` appliance model and `DR`, `HA` or `HA+DR` redundancy architecture):

```
Beginning phase 1 - Prepare this system to as the conversion source


This system has been detected as running on OL7. This will be the
PRIMARY
system that will provide the data to be replicated to a new OL8
system.
Your system has been detected as the following:
  Appliance Model: All-in-One
  Redundancy Architecture: DR
Is this correct? (y/n)
```

The script performs several pre-checks on the system to determine if there are known issues that will interfere with the conversion process. If any issues are encountered, the script will exit.

5. **For an HA+DR configuration only**: Choose which node type will be disconnected from DRBD and replaced with the Secondary (OL8) system. Selecting **DR** means you will be replacing your current Disaster Recovery node, while selecting **HA** means you will replace the peer HA node. ScienceLogic recommends that you replace the HA node, as this will allow reuse of the same IP addresses.

6. Enter the number listed next to the IP address on this system that will be used for the DRBD replication traffic.

7. Enter the IP address for the DRBD traffic to the new OL8 system.

8. The system will ask if you are ready to reconfigure Distributed Replication Block Device (DRBD) for conversion. Answer **yes** to proceed and answer **no** if you wish to exit. No outage is expected during this process, but the selected replacement node will be disconnected.

> **NOTE:** The cluster will be put into maintenance mode and will be unable to automatically fail over during the conversion process. DRBD replication will continue to work for the node that is not being replaced.

9. Wait as the system is reconfigured. When it does, Phase 1 of the conversion is complete. The system should have remained running with no interruption.

Phase 2: Prepare the Secondary System for Data Replication

Run the conversion script:

1. SSH in to the Secondary (Oracle Linux 8) system and elevate to root:

```
sudo -s
```

2. Run the conversion script:

```
silo-migrate
```

3. Enter an IP address for the Primary (Oracle Linux 7) system that is reachable by this system. This will connect to MariaDB and retrieve the conversion settings.

> **NOTE:** If the username and password in the local **silo.conf** cannot be used to connect to the Primary system, you will be prompted again for the IP address and then a username and password. This username and password are for connecting to MariaDB. After a connection is established, the local **silo.conf** will be updated with the correct information to authenticate automatically.

After it is connected, the script will retrieve the required information about the conversion, and prompt to shut down services on this system to reconfigure it to receive the data.

4. Confirm that you are ready to have the system services shut down and reconfigured for conversion. Type **y** to proceed and answer **n** if you wish to exit. If you type **y**, the system will reconfigure all the data to be migrated with DRBD.

Phase 2 of the conversion is complete, and data will start replicating. The speed of the replication will depend on system utilization and network bandwidth.

> **CAUTION:** Systems should not be rebooted or restarted while the conversion procedure is being executed. If the system is rebooted, it may take manual intervention to resume the synchronization process.

Phase 3: Wait for Data Replication to Finish

While waiting for the data replication process to finish, you can check the status by running the following commands:

- **For HA or DR configurations**: On the Primary (Oracle Linux 7) system when replacing the DR node:

```
drbdadm status
```

- **For HA, DR and HA+DR configurations**: On the Secondary (Oracle Linux 8) system:

```
drbdadm status
```

- **For HA+DR only**: On the Primary (Oracle Linux 7) system when replacing the DR node:

```
drbdadm status -S
```

- **For HA+DR only**: On the Primary (Oracle Linux 7) system when replacing the HA node:

```
drbdadm status
```

Review the output to see the percentage completed (the `done` value):

- **For HA, DR, and HA+DR configurations**: On the Primary (Oracle Linux 7) system when replacing the DR node:)

```
r0 role:Primary
  disk:UpToDate
  peer role:Secondary congested:yes
    replication:SyncSource peer-disk:Inconsistent done:81.25
```

- **For HA, DR, and HA+DR configurations**: On the Secondary (Oracle Linux 8): system:

```
r0 role:Secondary
  disk:Inconsistent
  migration_ol7 role:Primary
    replication:SyncTarget peer-disk:UpToDate done:83.68
```

- **For HA+DR configurations**: On the Primary (Oracle Linux 7) system when replacing the HA node:

```
r0-L role:Primary
  disk:UpToDate
  peer role:Secondary congested:yes
    replication:SyncSource peer-disk:Inconsistent done:81.25
```

When the system is fully replicated and in sync, the script will indicate that the peer-disk is up to date:

- **For HA, DR, and HA+DR configurations**: On the Primary (Oracle Linux 7) system when replacing the DR node:

```
r0 role:Primary
  disk:UpToDate
  peer role:Secondary
    replication:Established peer-disk:UpToDate
```

- **For HA, DR, and HA+DR configurations**: On the Secondary (Oracle Linux 8): system:

```
r0 role:Secondary
  disk:UpToDate
  migration_ol7 role:Primary
    peer-disk:UpToDate
```

- **For HA+DR configurations**: On the Primary (Oracle Linux 7) system when replacing the HA node:

```
r0-L role:Primary
  disk:UpToDate
  peer role:Secondary
    replication:Established peer-disk:UpToDate
```

Phase 4: Gracefully Demote the Primary (Source) System

Run the conversion script:

1. SSH in to the Primary (Oracle Linux 7) system and elevate to root:

    ```
    sudo -s
    ```

2. Run the conversion script:

    ```
    silo-migrate
    ```

3. The system will be checked for replication status, and you will be prompted to proceed with halting SL1 services on the system. Type **y** to proceed, and the system will gracefully halt SL1 services. Type **n** if you wish to exit.

4. Wait for the system to halt. When it does, Phase 4 of the conversion is complete.

Phase 5: Gracefully Promote the Secondary (Destination) System

Run the conversion script:

1. SSH in to the Secondary (Oracle Linux 8) system and elevate to root:

    ```
    sudo -s
    ```

2. Run the conversion script:

    ```
    silo-migrate
    ```

3. The system will be checked for replication status, and you will be prompted to proceed with bringing the system online. Type **y** to proceed, and the system will reconfigure and start all SL1 services. Type **n** if you wish to exit.

4. Wait for the system to restart and be accessible. When it does, Phase 5 of the conversion is complete.

Validation

Ensure the conversion was successful and the SL1 system is functioning by running the tasks in *After You Convert to OL8*.

Additional Configuration

As needed, see the following topics to convert collectors and other ScienceLogic products to OL8:

- *Convert the Data Collectors and Message Collectors to Oracle Linux 8*. You should also *re-ISO the Administration Portal for your system* if you have not yet done so.

- *Convert PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.

- *Converting Restorepoint to Oracle Linux 8 (OL8)* in the Restorepoint documentation.

Customer-hosted Cloud Platform as a Service (PaaS)

The conversion process for a cloud Platform as a Service (PaaS) is for ScienceLogic customers who self-host SL1 in their own AWS account. The primary difference between a PaaS (customer-hosted) and a SaaS (ScienceLogic-hosted) configuration is that self-hosted customer deployments use an AWX jump host instead of Terraform.

Converting from Oracle Linux 7 to Oracle Linux 8 for an AWX distributed deployment takes a blue/green approach by recreating each Data Engine as an Oracle Linux 8 instance. For more information about blue/green deployments, see https://docs.aws.amazon.com/whitepapers/latest/overview-deployment-options/bluegreen-deployments.html.

Conversion Steps

> **IMPORTANT:** Before converting to Oracle Linux 8, you will first need to upgrade to SL1 version 12.1.x. For more information, see *Upgrade SL1 to Version 12.1.2*.

The Oracle Linux 8 conversion process for a cloud Platform as a Service (PaaS) instance includes the following steps:

1. *Review the prerequisites*.
2. *Step 1: Re-provision the passive Data Engine into an OL8 instance running 12.1.2*.
3. *Step 2: Re-add instances back to the load balancer*.
4. *Step 3: Configure and validate the new Data Engine*.
5. *Step 4: Repeat the steps for the now-passive Data Engine*.
6. *Step 5: Stop and then terminate previous DE EC2 instances*.
7. *Validation: Ensure the conversion was successful and the system is functioning*.

The following image gives a high-level overview of the conversion process for this SL1 configuration type:
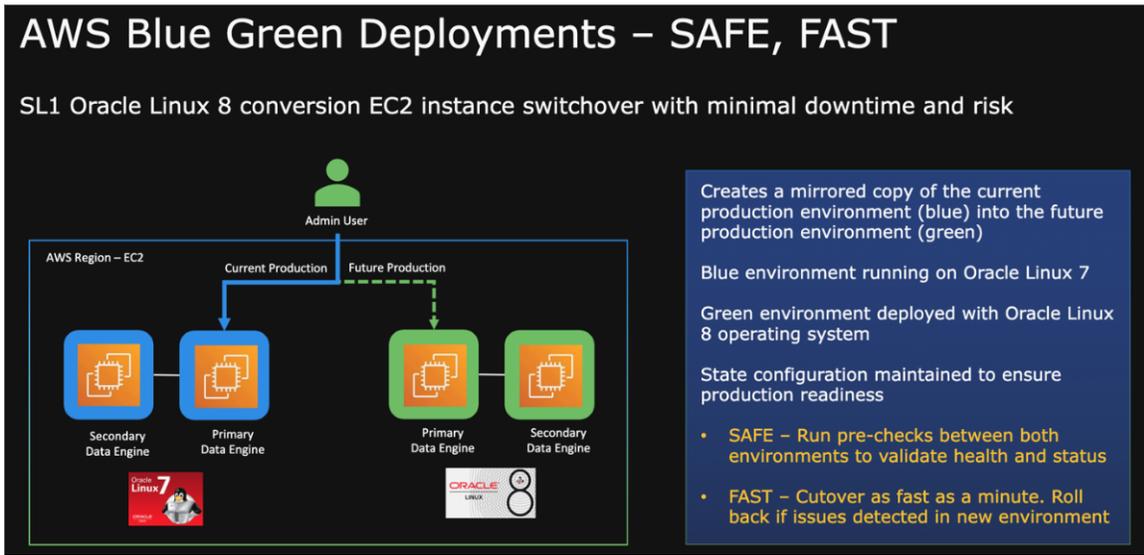
Prerequisites for Customer-hosted Cloud Platform as a Service (PaaS)

> **IMPORTANT:** ScienceLogic recommends that you read through this entire chapter before starting the conversion. Contact your ScienceLogic representative for assistance or more detailed guidance, if needed.

- A distributed stack running SL1 version 11.x that was deployed with the AWX framework.
- General AWS knowledge.

Step 1: Re-provision Passive Data Engine into an OL8 Instance Running SL1 12.1.2

1. Log in to the AWX instance.

2. Go to **templates** and select the **DE_Blue_Green** workflow template.

3. In the template, populate the inventory host below that corresponds to the *passive* Data Engine:
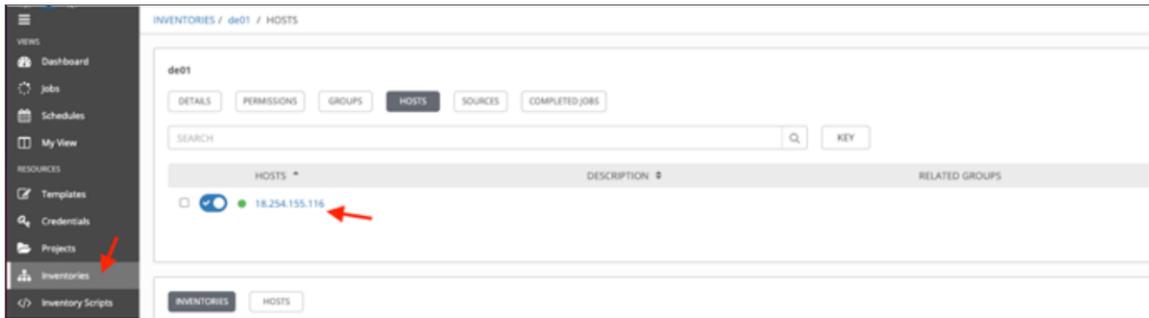
4. Find the Data Engine to IP association by using SSH to connect to the IP address.

5. Check the HA status:

   - `xxx@silo1717 ~ % ssh em7admin@18.254.155.1164. [em7admin@ip-10-249-2-113 ~]$ sudo /opt/em7/backend/ha_status.pypassive`

6. Set the passive Data Engine in the Job Settings:

7. Populate the **Extra Variables**, below:

```
---region: #AWS govcloud region client: #client name of AWX
deployment ol8_de_ami: ami-01d87d756095d3e7f #12.1.x OL8 AMI IDphone_
home: #true or falseconnectivity_type: #connectivity_type options:
public_ip, private_ip, or EIP
```

8. **Connectivity_Type**: Check to see how the Data Engine network is configured. Depending on customer configuration, there could be 1 of 3 different network configurations:

   - Data Engine with Elastic IP (EIP)

   - Data Engine with only a public IP

   - Data Engine with only a private IP (If the Data Engines are deployed with only a private IP, then the AWX instance will need to be deployed in on of the same subnets as the Data Engines.)

9. **EIP Option**: If the customer is deployed with an EIP and still has a jump host running, you can use the jumphost to re-add the existing EIP onto the new DE instance. If customer has no jump host then you will have to manually remove and re-add the EIP to the new instance:


10. Click **[Save]** and then launch.

Step 2: Re-add Instances Back to Load Balancer

This is an optional step that should be performed only if the customer is using a load balancer.

NOTE: When upgrading a data engine from Oracle Linux 7 (OL7) to OL8, a new precheck step verifies that load balancers and target groups are defined as expected to ensure that playbook executions pass successfully.

---

1. Login to the AWS console user interface.

2. Select **EC2 > Target Groups**.

3. Select the Target group associated with the customer:
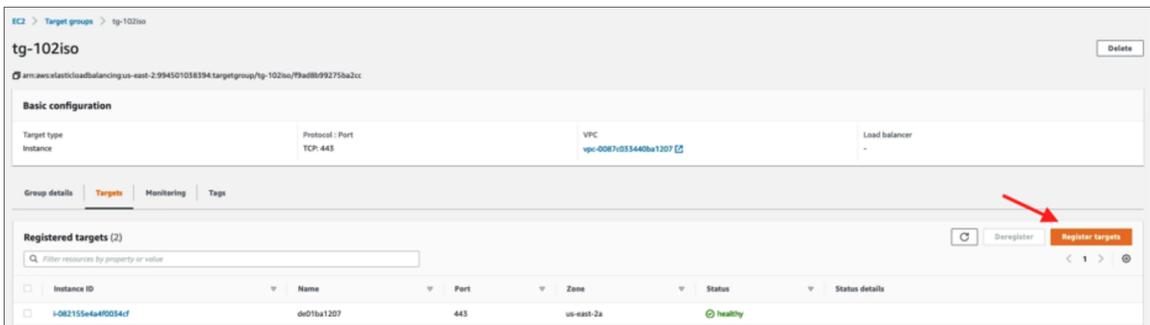
4. Click **[Register targets]**.

5. Add the new Data Engine to the target group:
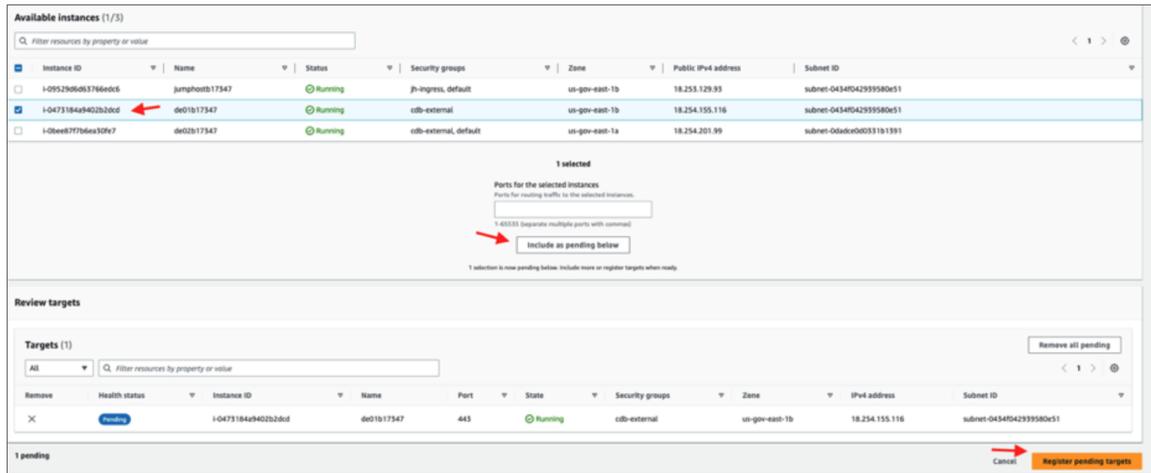
Step 3: Configure and Validate New Data Engine

1. After the job from Step 3 finishes, log in to the AWS console and make a note of the public and private IP of the new Data Engine. If the customer is using an EIP, then it will be the same IP of the previous passive Data Engine node.

2. As a basic health check, use the Data Engine IP to access the user interface:

    ```
    open browser2: https://<Data_Engine_IP>/em7/
    ```

3. SSH to the new Data Engine and make sure it is licensed:

    ```
    xxx@silo1717 ~ % ssh em7admin@18.254.155.116

    [em7admin@ip-10-249-2-113 ~]$ sudo licensed_stateAppliance is
    licensed
    ```

4. Determine the appliance ID by using the following command, and then set that appliance ID as active:

    ```
    [root@ip-10-249-2-113 em7]# silo_mysql -e "select id from
    master.system_settings_licenses where function=2 and name like
    '$(hostname)'"+----+| id |+----+| 13 |+----+[root@ip-10-249-2-113
    em7]# silo_mysql -e "update sysinfo.system_heartbeat set appliance_id
    = 13"
    ```

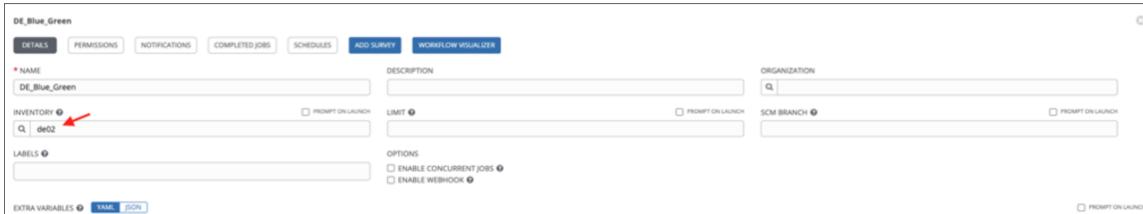5. Wait six minutes, and then check to see if the Data Engine is now active.

    ```
    [root@ip-10-249-2-113 em7]# sudo /opt/em7/backend/ha_status.pyActive
    ```

Step 4: Repeat the Steps for the Now-passive Data Engine

Repeat the same steps from Step 3, but now reference the job to the previously active Data Engine(which is now passive).

IE: DE02

Step 5: Stop and Then Terminate Previous Data Engine EC2 Instances

Depending on your level of comfort, you can stop and then terminate the EC2 instances. For production stacks, ScienceLogic recommends that you stop the Data Engine instance, and then wait 24 hours before terminating.

Validation

Ensure the conversion was successful and the SL1 system is functioning by running the tasks in *After You Convert to OL8*.

Additional Configuration

As needed, see the following topics to convert collectors and other ScienceLogic products to OL8:

- *Convert the Data Collectors and Message Collectors to Oracle Linux 8*. You should also *re-ISO the Administration Portal for your system* if you have not yet done so.
- *Convert PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.
- *Converting Restorepoint to Oracle Linux 8 (OL8)* in the Restorepoint documentation.

ScienceLogic-hosted Cloud Software as a Service (SaaS)

> **IMPORTANT:** For a *ScienceLogic-hosted cloud Software as a Service (SaaS)* configuration type, contact your ScienceLogic SRE or CSM to plan your OL8 conversion.

The SRE team will coordinate with SaaS customers to schedule the OL8 migration maintenance period. SaaS customers will be required to plan their collector conversion with the Customer Service Managers (CSM) so that the SRE team can plan for the collector conversion period.

The ScienceLogic SRE team is responsible for coordinating and converting the AWS SaaS components. The customer is responsible for converting the edge collectors. The SRE team will be in contact with each SaaS customer to coordinate the conversion date and time.

The conversion to Oracle Linux 8 for SaaS customers will include very minimal downtime.

Extended Architecture (AWS)

This configuration type includes an Extended Architecture (EA) Storage Node and a Storage Manager (Scylla) upgrade. This configuration type is only for Amazon Web Services (AWS); for customer on-premises see *Extended Architecture (on-premises)*. Also, a Compute Node managed by AWS EKS does not require a conversion to Oracle Linux 8.

This chapter covers the steps to upgrade any AWS 11.x deployment to 12.1.1 that includes both SL1 Distributed and Extended Components. This upgrade uses Terraform. The OL7 to OL8 upgrade must take place in the 12.1.x line before you can upgrade to 12.2.0.

> IMPORTANT: ScienceLogic recommends that you read through this entire chapter before starting the conversion. Contact your ScienceLogic representative for assistance or more detailed guidance, if needed.

Conversion Steps

1. *Step 1: Upgrade to SL1 12.1.2*. The operating system will still be OL7, and the Remote Database Server is still Aurora 2.

2. *Step 2: Upgrade the Extended Architecture to SL1 12.1.2*. This includes upgrading Scylla and EKS. The Storage Node and Storage Manager (Scylla) are upgraded to OL8, and EKS is upgraded to 1.27.

3. *Step 3: Upgrade the Data Engines to OL8*.

4. *Validation: Ensure the conversion was successful and the system is functioning*.

> NOTE: There can be a pause between any step, or they can be done without a pause. Also, you can do Step 3 at any time after step 1.

Step 1: Upgrade to SL1 12.1.2

The currently supported versions of SL1 running OL7 are versions 11.2, 11.3, and 12.1. If you are running a supported version of SL1 that is older than version 12.1.2, you will need to upgrade SL1 as part of the standard system update procedure, which is also known as the "silo_update" patch process.

> IMPORTANT: You must first *import* version 12.1.0 to your SL1 system and then upgrade to 12.1.1 before you can begin the Oracle Linux 8 operating system conversion.

For detailed information about updating SL1, see *Updating SL1* in the SL1 Product Documentation.

After this step, the operating system will still be OL7, and the Remote Database Server is still Aurora 2. Go to Step 2, below, to continue the conversion.

Step 2: Upgrade the Extended Architecture to SL1 12.1.1

In this step, the Storage Node (SN) and Storage Manager (Scylla) are upgraded to OL8, and EKS is upgraded to 1.27. ScienceLogic recommends that you do steps 1 and 2 together. You can split up these steps for larger customers to give more time for Scylla data sync.

Pre-upgrade steps:

1. Use SSH to access the Bastion host instance. Use the PEM key if necessary.

2. In the Management Node, navigate to the **sl1x-deploy** directory by running the following command at the shell prompt:

    ```
    cd /home/ec2-user/sl1x-deploy
    ```

3. Back up the following files:

    /home/ec2-user/sl1x-deploy/sl1x-inv.yml

    /home/ec2-user/sl1x-deploy/output-files/kube_config_cluster.yml

    > **TIP:** ScienceLogic recommends that you back up these files at regular intervals.

4. Delete any failed charts by running the following command:

    ```
    helm ls | awk '/FAILED/'
    ```

5. If the above command results in any output, run the following command:

    ```
    helm delete $(helm ls | awk '/FAILED/ { print $1 }')
    ```

6. Log in to the Harbor repository (if you are already logged in, you can skip this step):

    ```
    oras login registry.scilo.tools/sciencelogic/
    ```

    > **TIP:** To locate your credentials (username and password), log in to Harbor and click **User profile** in the top right corner. **CLI Secret** is your password.

7. Exit from **sl1x-deploy** and run the following commands to download the deployment files:

```
cd /home/ec2-user
```

```
oras pull registry.scilo.tools/sciencelogic/sl1x-deploy-tf:12.1.1
```

```
cd sl1x-deploy
```

8. Edit the **sl1x-inv.yml** file to match your SL1 Extended system. At the shell prompt, run the following command:

```
vi sl1x-inv.yml
```

9. Change the `sl1_version` value from `sl1_version:11.x.x` to `sl1_version:12.1.1`.

10. Add load balancer variables if they are not already there:

```
loadbalancer_type: clb
loadbalancer_enabled: true
loadbalancer_scheme: internet-facing
```

Upgrade Storage Manager (Scylla)

> **IMPORTANT:** Before upgrading the Storage Manager, you will need to scale down the streamer service so that agents can collect and keep data locally until the upgrade process finishes. To scale down, SSH to bastion and run the following command: `kubectl scale --replicas=0 deployment.apps/streamer`

You have two options for upgrading Storage Manager (Scylla):

- *Option 1: Backup and restore using an S3 bucket*
- *Option 2: OL8 blue/green redeploy using a secondary Data Center*

Option 1: Backup and Restore using an S3 Bucket

Prerequisites:

- Scylla S3 bucket
- Active Scylla cluster
- tfstate of previous deployment

> **NOTE:** Be sure to add the IAM Role referenced in the following link:
> https://manager.docs.scylladb.com/stable/backup/setup-aws-s3.html.

The following image gives a high-level overview of this option for the Storage Manager (Scylla) upgrade:
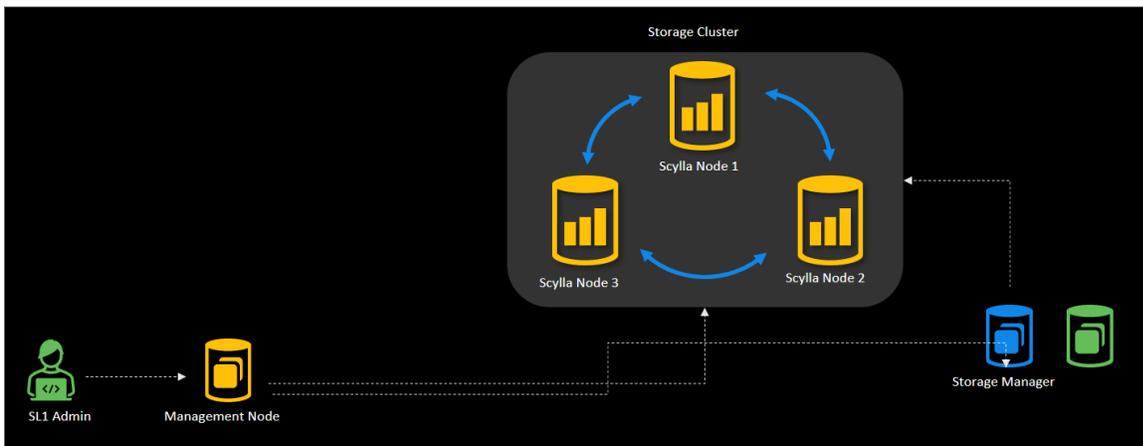
Backup and restore using an S3 bucket:

1. Edit **sl1x-inv.yml** to include variables for the S3 bucket:

    ```
    scylla_backup_bucket: scilo-scylla-backup
    ```

    ```
    scylla_backup_bucket_region: scilo-scylla-backup
    ```

    ```
    access_key: #######
    ```

    ```
    secret_key : #######
    ```

2. Run the following commands to back up Scylla data:

    ```
    cd /home/ec2-user/
    ```

    ```
    docker-compose -f docker-compose.external.yml run -- rm deploy
    backup-scylla-ol8
    ```

3. During execution of the command, take note of the output for the task:

```
TASK [sciencelogic.sl1x_sn.sn-scylla : Output Host IDs]
************************************************
changed: [10.152.1.250]


TASK [sciencelogic.sl1x_sn.sn-scylla : debug]
***********************************************************
ok: [10.152.1.250] => {
    "host_ids.stdout_lines": [
        "Datacenter: dc",
        "==============",
        "Status=Up/Down",
        "|/ State=Normal/Leaving/Joining/Moving",
        "--  Address       Load       Tokens      Owns     Host ID
                          Rack",
        "UN   10.152.5.250  9.05 MB    256         ?        a6a4758a-
5eb4-4382-99fb-b30e8841e68c  r2",
        "UN   10.152.3.250  9.09 MB    256         ?        d73d1ebb-
acdb-47ad-81dc-b675a1ac5234  r1",
        "UN   10.152.1.250  9.08 MB    256         ?        10de9ae4-
4c39-42c2-9ee0-6864244a4240  r0",
        "",
        "Note: Non-system keyspaces don't have the same replication
settings, effective ownership
information is meaningless"
    ]
}
```

4. SSH to the first Storage Node, `ssh ec2-user@10.152.1.250`, and get the snapshot tag:

```
[ec2-user@ip-10-152-1-250 ~]$ scylla-manager-agent download-files -L
s3:scilo-scylla-backup --list-snapshots

sm_20230214123551UTC
```

5. Upgrade the SN/SM to OL8 using Terraform. In the **terraform-aws-sl1x-paas** repository, check out the **release/12.1.1** branch and execute the following command:

```
terraform state rm module.eks[0].kubernetes_config_map.aws_auth[0]
```

6. Based on your `.tfvars` environment file, execute the following command to recreate SN/SM with OL8 and execute SN and SM playbooks:

```
bash run.sh -p aws -e 55acex113 -m 'module.storage_node[0]' module
```

7. Add the following variables in **sl1x-inv.yml**, based on the data from steps 3 and 4:

```
all:
  vars:
    #scylla backup and restore config
    scylla_backup_bucket: scilo-scylla-backup
    scylla_backup_bucket_region: us-east-1
    access_key: ************
    secret_key: ************
    # snapshot_tag specifies the Scylla Manager snapshot tag you want
to restore.
    snapshot_tag: sm_20230214123551UTC
    # host_id specifies a mapping from the clone cluster node IP to
the source cluster host IDs.
    # cluster host IDs.
    host_id:
      10.152.1.250: 10de9ae4-4c39-42c2-9ee0-6864244a4240
      10.152.3.250: d73d1ebb-acdb-47ad-81dc-b675a1ac5234
      10.152.5.250: a6a4758a-5eb4-4382-99fb-b30e8841e68c
```

8. Restore the playbook:

```
docker-compose -f docker-compose.external.yml run --rm deploy
restore-scylla-ol8
```

Option 2: OL8 Blue/Green Redeploy Using a Secondary Data Center

Upgrading the Oracle Linux operating system for the Storage Node from OL7 to OL8 is a three-step process that provides engineers with more control over the upgrade process. Breaking the upgrade process into these three steps lets engineer roll back or pause the upgrade for testing or scheduling purposes.

The three steps are:

1. **Upgrade**: This step is handled by the **snupgrade.yml** playbook, which creates and configures a three-node Scylla DR data center.

2. **Failover**: This step is handled by the **snfailover.yml** playbook, which fails over to the three-node DR created in the previous step.

3. **Post Failover**: This step is handled by the **snpostfailover.yml** playbook, which deletes the old Scylla data center.

The following image gives a high-level overview of this conversion option for the Storage Manager (Scylla) upgrade:
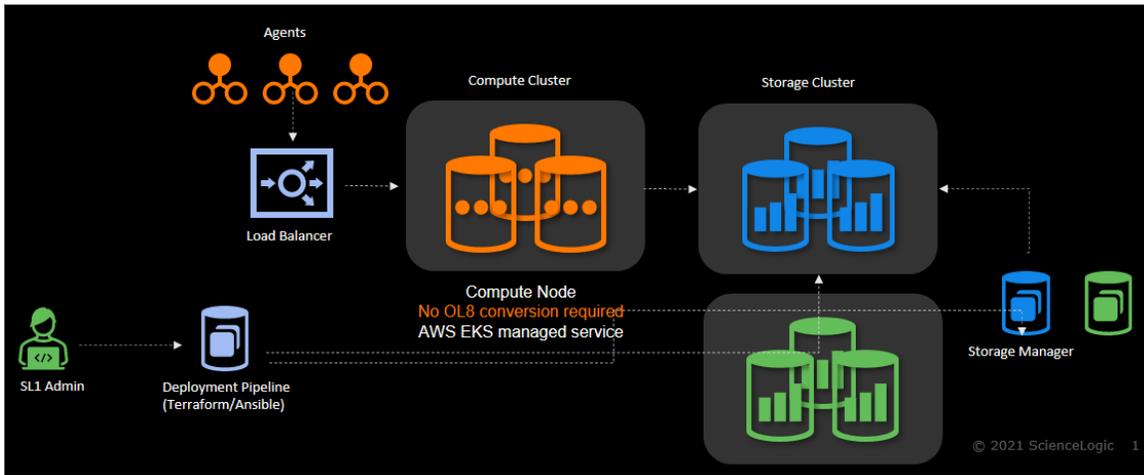
Step 1: Run a Storage Node upgrade

In this step, you will:

- Create 3 EC2 instances identical to the existing EC2 Instances for the Storage Node.

- Install Scylla and configure it.

- Set up replication.

> **NOTE:** For AMIID, look for `OL8.6-x86_64-HVM-2022-05-19` AMI ID in the AWS console.

```
docker-compose -f docker-compose.external.yml run --rm -e
REGION='<CLUSTER_REGION>' -e CLUSTER='<CLUSTER_NAME>' -e AMIID='<OL8_
AMI_ID>' deploy snupgrade
```

Step 2: Run failover

> **TIP:** Because replication might take some time to complete, you should do any necessary testing before moving to this step.

In this step, you will configure replication settings and fail over:

1. SSH to the Bastion Server and `cd sl1x-deploy directory`.

2. Run the following command:

    ```
    docker-compose -f docker-compose.external.yml run --rm deploy
    snfailover
    ```

3. When the command is complete, validate your data and make sure everything works as expected.

4. Go to the next step.

Step 3. Remove OL7 Storage Nodes

In this step, you will clean up the older Storage Nodes. This step permanently removes the old nodes. You do not have to run this step immediately after the failover.

```
docker-compose -f docker-compose.external.yml run --rm deploy
snpostfailover
```

Step 4. Update Storage Node hosts and update variables

In this step, you will updare the Storage Node hosts in **sl1x-inv.yml** and add a new Scylla cluster data center variable:

```
scylla_datacenter: dc2
```

NOTE: You can get new Storage Node hosts in AWS console and get private IP.

```
all:
  vars:
    scylla_datacenter: dc2


sn:
  hosts:
    10.152.1.X:
      ansible_user: ec2-user
    10.152.3.X:
      ansible_user: ec2-user
    10.152.5.X:
      ansible_user: ec2-user
```

Step 5. Run the Storage Node playbook

```
docker-compose -f docker-compose.external.yml run --rm deploy sn
```

Step 6. Update Cassandra service and charts

1. Update **sl1-cass** endpoint to the new private IPs for the Storage Nodes:

```
$ kubectl edit endpoints sl1-cass
.
.
subsets:
- addresses:
  - ip: 10.152.1.X
  - ip: 10.152.3.X
  - ip: 10.152.5.X
```

2. Verify the new endpoint:

```
$ kubectl get endpoints sl1-cass
NAME         ENDPOINTS                                          AGE
sl1-cass    10.152.1.231:9042,10.152.3.169:9042,10.152.5.24:9042   26h
```

3. Create and run the following script:

> **NOTE:** The first line of code below was broken after `"CASSANDRA_USER")]}` for readability in this
> document. You will need to remove the line break in your script.

```
for deploy in  $(kubectl get pods --all-namespaces  -o=jsonpath='
{range .items[*]}{.spec.containers[*].env[?(@.name=="CASSANDRA_
USER")]}
{"\t"}{.metadata.labels.app}{"\n"}{end}' | grep CASSANDRA | cut -f 2
| sort | uniq  )
do
    echo $deploy
    kubectl set env deployment/$deploy CASSANDRA_LOCAL_DATACENTER=dc2
done
```

4. Verify that all pods are running normally by running the following command:

```
kubectl get pods
```

Upgrading Scylla Manager

1. Run the following command to perform the Scylla Manager upgrade:

```
docker-compose -f docker-compose.external.yml run --rm deploy
smupgrade
```

2. Run the Scylla Manager playbook as normal:

```
docker-compose -f docker-compose.external.yml run --rm deploy sm
```

3. Validate that the deployments finished without issues, and make sure that Scylla Manager OL7 was removed and Scylla Manager OL8 was deployed.

Enable the Streamer Service

After any of the Storage Node and Scylla Manager upgrade procedures, you can scale up the streamer service again by running the following command:

```
kubectl scale --replicas=3 deployment.apps/streamer
```

Upgrade EKS

1. Use SSH to access the Bastion host instance. Use the PEM key, if needed.

2. In the Management Node, navigate to the **sl1x-deploy** directory by running the following command at the shell prompt to upgrade EKS 1.2x to 1.27:

```
docker-compose -f docker-compose.external.yml run --rm deploy
eksupgrade --tags eksprecheckdocker-compose -f

docker-compose.external.yml run --rm deploy eksupgrade
```

3. Delete the following service:

```
helm delete external-dns -n external-dns
```

4. To update the SL1 Extended system, run the deploy commands at the shell prompt:

```
docker-compose -f docker-compose.external.yml run --rm deploy paas
```

> **NOTE:** Service teams need to update their charts to take the `cassandra_datacenter_name` variable so that you can change it to the new `dc2`. Until then, execute the command to update charts version and move on to the workaround, below, to update new charts with `dc2`.

Workaround for upgrading EKS

1. Create and run the following script:

> **NOTE:** The first line of code below was broken after `"CASSANDRA_USER")]}` for readability in this document. You will need to remove the line break in your script.

```
for deploy in  $(kubectl get pods --all-namespaces  -o=jsonpath='
{range .items[*]}{.spec.containers[*].env[?(@.name=="CASSANDRA_
USER")]}
{"\t"}{.metadata.labels.app}{"\n"}{end}' | grep CASSANDRA | cut -f 2
| sort | uniq  )
do
    echo $deploy
    kubectl set env deployment/$deploy CASSANDRA_LOCAL_DATACENTER=dc2
done
```

2. Verify all pods are running normally by running the following command:

```
kubectl get pods
```

Step 3: Upgrade the Data Engines to OL8

> **TIP:** There can be a pause between any of the three main Steps, or they can be done without a pause. Also, you can do this Step at any time after *Step 1*.

> **NOTE:** The following steps are for terraform deployments, and a bastion host is required.

Before you start the upgrade, gather information about the Amazon Machine Image (AMI) and the SL1 product update file (**siloupdate**).

Both files should have the same version of SL1 and the same build, such as:

*   AMI: em7_x86_64_**12.1.0-3022**.el8-DB-Remote
*   Siloupdate: EM7_v**12.1.0.el7-3022**.Platform_2023-07-07.qa.siloupdate

    Note the version of **12.1.0** and build of **3022** on both file names.

To upgrade SL1 and convert to OL8:

1. Update the SL1 appliances to 12.1.x using the **siloupdate** file.

2. To verify that all appliances are updated to 12.1.x, connect to each appliance and run the following command on each appliance:

```
cat /etc/em7-release
```

3. SSH to one of the Data Engines and run the following command to set the cluster into maintenance:

```
silo_mysql -e "update sysinfo.system_heartbeat set maintenance = 1"
```

4. SSH to the Data Engines and check if the Data Engine is passive:

```
/opt/em7/backend/ha_status.py
```

5. Run the following command in the bastion for the passive node:

```
docker-compose -f docker-compose.external.yml run --rm \
-e REGION='<REGION>' \
-e CLUSTER='<CLUSTER_NAME>' \
-e COMPONENT='de02' \
-e AMI='<OL8_12.1_AMI_ID>' \
deploy deupgrade
```

> **NOTE:** In the above example, `de02` is a passive node. Only run these commands on the *passive* node, not the active node.

where:

`-e REGION='<REGION>'` is the region of your cluster.

`-e CLUSTER='<CLUSTER_NAME>'` is the cluster name.

`-e COMPONENT='de02'` is the component (either `de01` or `de02`, whichever is passive).

`-e AMI='<OL8_12.1_AMI_ID>'` is the AMI ID. To find the ID, go to AWS and search for the AMI that corresponds to the version and the build number of **siloupdate**. Copy the ID and provide it here.

> **NOTE:** Optionally, you can provide a specific private IP for the new Data Engine: `-e DEIP='<Private_IP_Address>'`.

These playbook steps will create a new node running Oracle Linux 8 and attach the EIP from the current passive node.

6. After you run these steps and make sure everything is working, make the new Oracle Linux 8 node an active node by connecting to it and running the following command:

    ```
    silo_mysql -e "update sysinfo.system_heartbeat set appliance_id = <DE
    appliance ID>"
    ```

    > **TIP:** You can find the `appliance ID` on the **Appliance Manager** page (System > Settings > Appliances) for the newly created node.

7. After running the above command, the newly created node is now the Active node, and the other Data Engine is the Passive node. You can now can run the upgrade on this node, but first you will need to make sure that everything is working as expected:

    *   Verify that the collectors are still available by checking the **Collector Status** page (System > Monitor > Collector Status) in the SL1 user interface.

    *   Make sure that all necessary services are up and running.

    *   Make sure you can access the SL1 user interface from this new node.

    *   Delete the old Oracle Linux 7 node from SL1 on the **Appliance Manager** page (System > Settings > Appliances) and then terminate the instance in AWS.

8. Repeat the steps above for the second Data Engine. In this example `de01` is now passive node:

    ```
    docker-compose -f docker-compose.external.yml run --rm \
    -e REGION='<REGION>' \
    -e CLUSTER='<CLUSTER_NAME>' \
    -e COMPONENT='de01' \
    -e AMI='<OL8_12.1_AMI_ID>' \
    deploy deupgrade
    ```

9. After both Data Engines are updated, SSH to one of the Data Engines and run the following command to remove the maintenance mode:

    ```
    silo_mysql -e "update sysinfo.system_heartbeat set maintenance = 0"
    ```

10. Update **sl1x-inv.yml** file with new private IPs for future upgrades:

```
ap:
  hosts:
    10.152.0.X:
      ansible_user: em7admin
    10.152.2.X:
      ansible_user: em7admin
cdb:
  hosts:
    10.152.0.X:
      ansible_user: em7admin
```

Validation

Ensure the conversion was successful and the SL1 system is functioning by running the tasks in *After You Convert to OL8*.

Additional Configuration

As needed, see the following topics to convert collectors and other ScienceLogic products to OL8:

- *Convert the Data Collectors and Message Collectors to Oracle Linux 8*. You should also *re-ISO the Administration Portal for your system* if you have not yet done so.
- *Convert PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.
- *Converting Restorepoint to Oracle Linux 8 (OL8)* in the Restorepoint documentation.

Extended Architecture (on-premises)

This configuration type includes an Extended Architecture (EA) Storage Node and a Storage Manager (Scylla) upgrade. This configuration type is only for customer on-premises Extended Architecture.

> IMPORTANT: ScienceLogic recommends that you read through this entire chapter before starting the conversion. Contact your ScienceLogic representative for assistance or more detailed guidance, if needed.

Conversion Steps

1. *Step 1: Upgrade to SL1 12.1.2* (triggers any schema changes).
2. *Step 2: Re-ISO the passive Database Server*, fail over, and re-ISO the newly passive Database Server.

3. *Step 3: Re-ISO or install new Data and Message Collectors*.

4. *Step 4: Upgrade the Extended Architecture to SL1 12.1.2*.

5. *Validation: Ensure the conversion was successful and the system is functioning*.

Step 1: Upgrade to SL1 12.1.2

The currently supported versions of SL1 running OL7 are versions 11.2, 11.3, 12.1. If you are running a supported version of SL1 that is older than version 12.1.2, you will need to upgrade SL1 as part of the standard system update procedure, which is also known as the "silo_update" patch process.

> IMPORTANT: You must first *import* version 12.1.0 to your SL1 system and then upgrade to 12.1.1 before you can begin the Oracle Linux 8 operating system conversion.

For detailed information about updating SL1, see *Updating SL1* in the SL1 Product Documentation.

Step 2: Re-ISO the Passive Database Server

A cluster includes an *active* Database Server and a *passive* Database Server. The passive Database Server provides redundancy and is dormant unless a failure occurs on the active Database Server. SL1 uses block-level replication to ensure that the data on each Database Server's primary file system is identical and that each Database Server is ready for failover if necessary. If the active Database Server fails, the passive Database Server automatically becomes active and performs all required database tasks. The previously passive Database Server remains active until another failure occurs.

For Step 2, you will need to update the passive Database Server:

1. Re-ISO the passive Database Server.

2. Perform a failover.

3. Re-ISO the newly passive Database Server.

Step 3: Re-ISO or install New Data and Message Collectors

For Step 3, you will need to perform an in-place upgrade of Data and Message Collectors to OL8. This process is covered in *Message Collectors and Data Collectors*.

Step 4: Upgrade the Extended Architecture to SL1 12.1.1

In this step, the following changes are made:

- The Storage Node (SN) is upgraded to 12.1.1 OL8 via a rolling upgrade.

- The Storage Manager (Scylla) is upgraded to 12.1.1 OL8 via re-ISO.

- The Compute Node (CN) is upgraded to 12.1.1 OL8 via backup, then re-ISO.

- The Load balancer (LB) is upgraded via 12.1.1 OL8 re-ISO.

- The Management Node (MN) is upgraded to 12.1.1 OL8 via re-ISO.

Pre-upgrade Steps

1. Use SSH to access the Management Node. Open a shell session on the server and log in with the system password you defined in the ISO menu.

2. In the Management Node, navigate to the **sl1x-deploy** directory by running the following command at the shell prompt:

    ```
    cd sl1x-deploy
    ```

3. Log in to the Harbor repository (if you are already logged in, you can skip this step):

    ```
    oras login registry.scilo.tools/sciencelogic/
    ```

    > **TIP:** To locate your credentials (username and password), log in to Harbor and click **User profile** in the top right corner. **CLI Secret** is your password.

4. Exit from **sl1x-deploy** and run the following commands to download the deployment files:

    ```
    cd /home/em7admin/
    ```

    ```
    oras pull registry.scilo.tools/sciencelogic/sl1x-deploy:12.1.1
    ```

    ```
    cd sl1x-deploy
    ```

5. Copy the inventory template file to the name **sl1x-inv.yml**:

    ```
    cp sl1x-inv-template.yml sl1x-inv.yml
    ```

6. Edit the **sl1x-inv.yml** file to match your SL1 Extended system. At the shell prompt, run the following command:

    ```
    vi sl1x-inv.yml
    ```

    > **CAUTION:** Do not remove colons when editing this file.

7. Make sure that the `sl1_version` value is set to `sl1_version:12.1.1`.

8. Supply values in all the fields that are applicable, and then save your changes and exit the file (`:wq`).

9. Pull the IAC Docker image according to the **docker-compose** file:

```
docker-compose -f docker-compose.external.yml pull
```

Upgrade Storage Manager (Scylla)

You have two options for upgrading Storage Manager (Scylla):

- *Option 1: Backup and restore using an AWS S3 bucket for smaller environments.*
- *Option 2: A rolling upgrade for typical environments.*

Option 1: Backup and Restore using an S3 Bucket

Prerequisites:

- Scylla S3 bucket
- Active Scylla cluster
- tfstate of previous deployment

> **NOTE:** Be sure to add the IAM Role referenced in the following link:
> https://manager.docs.scylladb.com/stable/backup/setup-aws-s3.html.

To back up and restore using an S3 bucket:

1. Disable the streamer service by scaling it down so that agents can collect and keep data locally until the SN/SM upgrade process finishes. To scale down, SSH to the Management Node (MN) and run the following command:

```
kubectl scale --replicas=0 deployment.apps/streamer
```

2. Exit the Ansible shell and edit **sl1x-inv.yml** to include variables for the S3 bucket:

```
scylla_backup_bucket: scilo-scylla-backup
```

```
scylla_backup_bucket_region: scilo-scylla-backup
```

```
access_key: #######
```

```
secret_key : #######
```

3. Run the following command to back up Scylla data:

```
cd /home/ec2-user
```

```
docker-compose -f docker-compose.external.yml run -- rm deploy
backup-scylla-ol8
```

4. During execution of the command, take note of the output for the task:

```
TASK [sciencelogic.sl1x_sn.sn-scylla : Output Host IDs]
*********************************************************************
changed: [10.152.1.250]


TASK [sciencelogic.sl1x_sn.sn-scylla : debug]
*********************************************************************
*********
ok: [10.152.1.250] => {
    "host_ids.stdout_lines": [
        "Datacenter: dc",
        "==============",
        "Status=Up/Down",
        "|/ State=Normal/Leaving/Joining/Moving",
        "--   Address        Load        Tokens        Owns     Host ID
                            Rack",
        "UN   10.152.5.250  9.05 MB     256           ?        a6a4758a-
5eb4-4382-99fb-b30e8841e68c  r2",
        "UN   10.152.3.250  9.09 MB     256           ?        d73d1ebb-
acdb-47ad-81dc-b675a1ac5234  r1",
        "UN   10.152.1.250  9.08 MB     256           ?        10de9ae4-
4c39-42c2-9ee0-6864244a4240  r0",
        "",
        "Note: Non-system keyspaces don't have the same replication
settings, effective ownership information is meaningless"
    ]
}
```

5. SSH to the first Storage Node, `ssh ec2-user@10.152.1.250`, and get the snapshot tag:

```
[ec2-user@ip-10-152-1-250 ~]$ scylla-manager-agent download-files -L
s3:scilo-scylla-backup --list-snapshots
```

```
sm_20230214123551UTC
```

6. Re-iso the SN/SM nodes with the 12.1.1 OL8 ISO:

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit sn,sm
```

7. SSH to the MN and finish SN/SM deployment by running the following commands:

```
cd /home/ec2-user/
```

```
docker-compose -f docker-compose.external.yml run --rm deploy sn
```

```
docker-compose -f docker-compose.external.yml run --rm deploy sm
```

8. Add the following variables in **sl1x-inv.yml**, based on the data from steps 3 and 4:

```
all:
  vars:
    #scylla backup and restore config
    scylla_backup_bucket: scilo-scylla-backup
    scylla_backup_bucket_region: us-east-1
    access_key: ************
    secret_key: ************
    # snapshot_tag specifies the Scylla Manager snapshot tag you want
to restore.
    snapshot_tag: sm_20230214123551UTC
    # host_id specifies a mapping from the clone cluster node IP to
the source cluster host IDs.
    # cluster host IDs.
    host_id:
      10.152.1.250: 10de9ae4-4c39-42c2-9ee0-6864244a4240
      10.152.3.250: d73d1ebb-acdb-47ad-81dc-b675a1ac5234
      10.152.5.250: a6a4758a-5eb4-4382-99fb-b30e8841e68c
```

9. Restore the playbook:

```
docker-compose -f docker-compose.external.yml run --rm deploy
restore-scylla-ol8
```

10. Enable the streamer service and scale it up again:

```
kubectl scale --replicas=3 deployment.apps/streamer
```

Option 2: Rolling upgrade for typical environments

To configure a rolling upgrade:

1. Remove the first Scylla node from the cluster by running the following command:

```
docker-compose -f docker-compose.external.yml run --rm deploy sn-
remove --limit sn[0]
```

2. Re-ISO the first Scylla node with OL8 ISO. These Scylla node IPs can be found in the sl1x-inv.yml file. For example:

```
sn:
  hosts:
    10.2.253.90: # ip of storage node 1
    10.2.253.91: # ip of storage node 2
    10.2.253.92: # ip of storage node 3
  vars:
    # roles/sn-scylla
    scylla_admin_username: em7admin # scylla admin username
    scylla_admin_password: em7admin # scylla admin password

sm:
  hosts:
    10.2.253.82: # ip of sm
```

3. Re-add the first Scylla node to the cluster:

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit sn[0]
```

```
docker-compose -f docker-compose.external.yml run --rm deploy sn-
restore --limit sn[0]
```

4. Check to make sure the node added correctly:

    ```
    docker-compose -f docker-compose.external.yml run --rm deploy sn-
    cluster-check --limit sn[0]
    ```

    If this playbook fails, then the new node has not joined the cluster yet. Wait 15 minutes for the node to join, and then re-run the playbook. For larger clusters you will need to wait more time, so you can continue checking via this method until you see success.

    A failure will look like the following (click the image to enlarge):

    A success will look like the following (click the image to enlarge):

5. Remove the second and third Scylla node from the cluster:

    ```
    docker-compose -f docker-compose.external.yml run --rm deploy sn-
    remove --limit sn[1]
    ```

    ```
    docker-compose -f docker-compose.external.yml run --rm deploy sn-
    remove --limit sn[2]
    ```

    NOTE: For large amounts of data, remove the nodes one by one (first the second node, and then the third node).

6. Re-ISO the second and third Scylla node with the 12.1.1 OL8 ISO.

7. Re-iso the SM node with the 12.1.1 OL8 ISO.

8. Re-add the second and third Scylla node to the cluster:

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit sn[1],sn[2]
```

```
docker-compose -f docker-compose.external.yml run --rm deploy sn-
restore --limit sn[1],sn[2]
```

9. Check to make sure the node added correctly:

```
docker-compose -f docker-compose.external.yml run --rm deploy sn-
cluster-check --limit sn[1]
```

```
docker-compose -f docker-compose.external.yml run --rm deploy sn-
cluster-check --limit sn[2]
```

10. Deploy the SM:

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit sm
```

```
docker-compose -f docker-compose.external.yml run --rm deploy sm
```

Upgrade the Compute Node to 1.27 running OL8

High level process:

1. Back up data to control plane nodes.
2. Re-ISO worker nodes.
3. Restore data to worker nodes.
4. Re-ISO control plane nodes.
5. Re-deploy the cluster.

There are separate processes for six-node and three-node clusters, and they are both outlined below.

> **NOTE:** If you do not want to back up Compute Node (CN) data, or if you do not have AI/ML data stored, you can simply re-ISO the nodes to OL8 ISO then run step 5 to re-deploy the cluster.

Six-node Compute Node Upgrade

1. Run the backup:

```
docker-compose -f docker-compose.external.yml run --rm deploy rke-
backup --tags 6+nodes
```

2. Re-ISO the CN worker nodes to 12.1.1 OL8 ISO. These worker node IPs can be found in the **sl1x-inv.yml** file. For example:

```
cn:
  children:
    #rke will setup all roles on the master nodes if no worker nodes
are present
    master:
      hosts:
          10.2.253.240: # ip of compute/master node 1
          10.2.253.241: # ip of compute/master node 2
          10.2.253.242: # ip of compute/master node 3
    worker:
      hosts:
          10.2.253.243:
          10.2.253.244:
          10.2.253.245:
```

3. Set up SSH-keys to the newly ISO'd worker nodes and restore the data:

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit worker
```

```
docker-compose -f docker-compose.external.yml run --rm deploy rke-
restore --tags 6+nodes
```

4. Re-ISO the CN master nodes to 12.1.1 OL8 ISO.

5. If configured, re-ISO the load balancers to 12.1.1 OL8 ISO.

6. Set up SSH-keys to the newly ISO'd master node and re-deploy the cluster:

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit master,lb
```

```
docker-compose -f docker-compose.external.yml run --rm deploy cn
```

```
docker-compose -f docker-compose.external.yml run --rm deploy app
```

Three-node Compute Node Upgrade

1. Run the backup:

```
docker-compose -f docker-compose.external.yml run --rm deploy rke-
backup --tags 3nodes
```

2. Re-ISO the first two master nodes in the inventory file. You can find these IPs in the **sl1x-inv.yml** file: 10.2.253.240 and 10.2.253.241

```
cn:
  children:
    #rke will setup all roles on the master nodes if no worker nodes
are present
    master:
      hosts:
          10.2.253.240: # ip of compute/master node 1
          10.2.253.241: # ip of compute/master node 2
          10.2.253.242: # ip of compute/master node 3
    worker:
      hosts:
          10.2.253.243:
          10.2.253.244:
          10.2.253.245:
```

3. Set up SSH-keys to the newly iso'd master nodes and restore the data:

```
echo > ~/.ssh/known_hosts
```

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit master[0],master[1]
```

```
docker-compose -f docker-compose.external.yml run --rm deploy rke-
restore --tags 3nodes
```

4. Re-ISO the last master node in the inventory file to OL8 ISO.

5. If configured, re-ISO the load balancers to 12.1.1 OL8 ISO.

6. Set up SSH-keys to the newly ISO'd master node and re-deploy the cluster:

```
echo > ~/.ssh/known_hosts
```

```
docker-compose -f docker-compose.external.yml run --rm deploy ssh-
keys --ask-pass --limit master[2],lb
```

```
docker-compose -f docker-compose.external.yml run --rm deploy cn
```

```
docker-compose -f docker-compose.external.yml run --rm deploy app
```

Upgrade the Management Node

> **NOTE**: Do not upgrade the Management Node (MN) until all SL1 appliances have been upgraded to OL8. This includes the Database Server (CDB), Administration Portal (AP), Load Balancer (LB), Compute Nodes (CN), Storage Nodes (SN), and the Storage Manager (SM) are upgraded to OL8.

1. Backup the **sl1x-deploy** directory and SSH to the Management Node (MN):

    ```
    cd /home/em7admin/
    ```

    ```
    tar cvf sl1x-deploy.tgz sl1x-deploy
    ```

2. Copy the compressed file to a secure machine. As an example, you can use the following command:

    ```
    scp em7admin@<MN_IP>:sl1x-deploy.tgz sl1x-deploy.tgz
    ```

3. Re-ISO the Management Node to 12.1.1 OL8 ISO.

4. Install Oras by SSHing to the MN and running the following commands to download and install Oras:

    ```
    sudo su
    ```

    ```
    curl -LO https://github.com/oras-
    project/oras/releases/download/v0.12.0/oras_0.12.0_linux_amd64.tar.gz
    ```

    ```
    tar -zxf oras_0.12.0_*.tar.gz -C /usr/bin/
    ```

    ```
    rm -rf oras_0.12.0_*.tar.gz
    ```

    ```
    exit
    ```

5. Log in to the Harbor repository (if you are already logged in, you can skip this step):

    ```
    oras login registry.scilo.tools/sciencelogic/
    ```

> **TIP:** To locate your credentials (username and password), log in to Harbor and click **User profile** in the top right corner. **CLI Secret** is your password.

6. Pull and run the **mn-transformation-script**:

```
oras pull registry.scilo.tools/sciencelogic/mn-transformation:MN-
Trans-OL8
```

```
mv mn-transformation.sh /tmp/
```

```
sudo sh /tmp/mn-transformation.sh
```

7. To apply the changes from the **mn-transformation.sh** script, exit the SSH session:

```
exit
```

8. Copy the compressed file back to the MN. As an example, you can use the following command:

```
scp sl1x-deploy.tgz em7admin@<MN_IP>:/home/em7admin/sl1x-deploy.tgz
```

9. SSH to your MN and restore the **sl1x-deploy** directory:

```
cd /home/em7admin/
```

```
tar xf sl1x-deploy.tgz -C ./
```

10. Your Management Node is now set up, and you can manage the cluster. To test it, you can run the below commands to see the kubectl pod output:

```
[em7admin@ntoor-mn-10-2-253-86 sl1x-deploy]$ docker-compose -f
docker-compose.external.yml run --rm deploy shell
```

```
INFO:__main__:Running with Parameters: Namespace(ansible_args=[],
command='shell', force_root=False)
```

```
ansible@74c0d0905aa7:/ansible$ kubectl get pods
```

```
NAME READY STATUS RESTARTS AGE
```

```
agent-log-alert-7658fb8bcc-d865d 1/1 Running 0 2d2h
```

```
agent-vitals-interface-alert-77cd79569b-ctnzx 1/1 Running 0 2d2h
```

```
agent-vitals-interface-alert-77cd79569b-dgx7k 1/1 Running 0 2d2h
```

Enable the Streamer Service

After you complete the upgrade procedures, you can scale up the streamer service again by running the following command:

```
kubectl scale --replicas=3 deployment.apps/streamer
```

Validation

Ensure the conversion was successful and the SL1 system is functioning by running the tasks in *After You Convert to OL8*.

Additional Configuration

As needed, see the following topics to convert collectors and other ScienceLogic products to OL8:

- *Convert the Data Collectors and Message Collectors to Oracle Linux 8*. You should also *re-ISO the Administration Portal for your system* if you have not yet done so.
- *Convert PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.
- *Converting Restorepoint to Oracle Linux 8 (OL8)* in the Restorepoint documentation.

Message Collectors and Data Collectors

Notes about the conversion process for SL1 Message Collectors and Data Collectors:

- Converting collectors from Oracle Linux 7 (OL7) to Oracle Linux 8 (OL8) involves an *in-place collector conversion*.

    - This conversion process requires SL1 12.1.2 and AP2 version 8.14.26 (Gelato).

    - Your OL7 collectors should be running MariaDB version 10.4.29. For more information, see *upgrading MariaDB to 10.4.29 for OL7 appliances*.

- Alternately, you can do a *redeploy (re-ISO) of the collectors* to convert them to OL8.
- If the in-place collector upgrade process fails, you can still re-ISO the collector to convert it to OL8.
- There are single or bulk upgrade options for the in-place collector conversion process; however, if you opt to redeploy (re-ISO) your collectors, that must be done individually.
- The conversion time depends on the number of collectors.

Conversion Methods

You have the choice of two methods for converting SL1Message Collectors and Data Collectors to Oracle Linux 8:

- *Option 1: Run the in-place collector conversion process*. If you need to keep the original collectors, you will need to run the in-place collector conversion process using the command-line interface and the SL1 user interface.
- *Option 2: Redeploy (re-ISO) existing collectors with new collectors*. If you consider the collector appliance as a "throwaway" appliance, or if you have very few collectors deployed, then you can replace (re-ISO) the existing collectors by deploying new 12.1.2 OL8 collectors instead of keeping the original collectors.

Option 1: Run the In-place Collector Conversion Process

If you need to keep your current set of SL1 Message Collectors and Data Collectors, you will need to run the OL8 collector conversion process from the command-line interface and on the **Collector Groups** page in SL1. This process is only available for SL1 version 12.1.2 or later, and AP2 version 8.14.26 (Gelato).

The OL8 collector conversion process is best suited for large collector networks for which you want to maintain the existing infrastructure. The process is designed to replace the existing collector appliance with the OL8 operating system and reapply all existing configurations. When the OL8 conversion is complete, the collector will reconnect to the database and resume data collection. You can convert one, many, or all at the same time.

Your OL7 collectors should be running MariaDB version 10.4.29. For more information, see *upgrading MariaDB to 10.4.29 for OL7 appliances*.

> **TIP:** ScienceLogic recommends that you run a test against a single collector first to get comfortable with the process. When you are ready to migrate the remaining collectors, you can select them in batches to complete the conversion process.

Overview

The collector conversion procedure executes a headless reinstallation while preserving system-specific data so that the collector can be contacted after the conversion is completed. This is accomplished by the following components:

- A distribution upgrade service that runs on the active Database Server/Data Engine. This service distributes the required data and run processes on the collectors to coordinate the conversion process.

- A script that runs on collectors that performs the conversion process and preserves specific data and does a headless reinstallation of the system.

The conversion process uses a Linux kernel feature called *kexec*, which allows a new kernel to be loaded in place of a current running kernel, and then switch operation to that kernel without interruption. Due to the wide variety of hardware, cloud, virtualization platforms and workloads on which you can deploy SL1, unexpected issues might arise during the switchover and execution of the new kernel. These issues can be difficult or impossible to detect prior to the redeployment process, and might lead to total failure. In this situation, you will need to do a re-ISO of the collector.

Guidelines for the Conversion

Review the following guidelines to prepare your SL1 systems and collectors for the conversion:

- When using virtualization systems, use the highest CPU compatibility mode available. In VMware this is referred to as EVC mode. ScienceLogic recommends that you use Broadwell or higher if available. Refer to your virtualization system vendor to understand CPU compatibility.

- Make sure all firmware is up to date. This includes BIOS, BMCs, and other related firmware. A majority of the functions and hardware initialization is done through ACPI calls. You should have the latest firmware to ensure the best compatibility. This is important to ensure your hosts are updated, even if your collectors are VMs. Refer to your vendors to ensure that you have the latest firmware applied to your systems.

- Ensure that your hardware or virtual machine is compatible with Oracle Linux 8. Hardware requirements have changed between OL7 and OL8. The running kernel is switched live from OL7 to OL8, which can result in a conversion failing.

- Ensure that all systems have sufficient resources. This is critical for virtualized systems to not be resource-constrained or over-committed during this process. Review your systems with your administrators and check that they are operating within vendor best practice guidelines.

Conversion Workflow

Complete the following steps to convert collectors to OL8:

1. *Step 1: Review the list of prerequisites.*

2. *Step 2: Upgrade all SL1 appliances to SL1 version 12.1.2*.

   - This step includes *upgrading MariaDB to 10.4.29 for OL7 appliances*.
   - This step includes *converting the Database Server or Data Engine to OL8* (which upgrades the Database Server or Data Engine to MariaDB 10.4.31).

3. *Step 3: Upgrade all SL1 appliances to AP2 version 8.14.26 (Gelato)*.

4. *Step 4: Use the command-line interface to stage the .iso file*.

5. *Step 5: Use the Collector Groups page in SL1 to convert collectors to OL8*.

Step 1: Review the List of Prerequisites

> **IMPORTANT:** The in-place conversion checks to make sure a collector has a minimum of 6 GB of RAM and 90 GB of hard disk space to operate. All collectors should always be sized to the guidelines on the ScienceLogic SL1 System Requirements page, which is typically above these minimum values. If the collector fails this minimum viable check, or if it does not meet the system requirements, you can optionally increase the disk size to the recommended size so you can use the conversion utility, or you can *re-ISO the collector* at the recommended guidelines.

> **IMPORTANT:** ScienceLogic also highly recommends that you enable failover for your collector groups that will be impacted by upgrading to 12.1.2 and converting to OL8. For more information on enabling failover, see the section on *Collector Groups* in the Product Documentation.

Make sure that your SL1 appliances meet the following prerequisites:

- You will need access to the ScienceLogic Support Portal at https://support.sciencelogic.com/s/ to download the 12.1.2 **.iso** file. You will need to access the sha512sum for the 12.1.2 **.iso** file, which is also available on the Support Portal when downloading the ISO file.

- All SL1 appliances must be upgraded to version 12.1.2, including MariaDB updates. Your OL7 collectors should be running MariaDB version 10.4.29. See *Step 2*, below.

- The Database Server or Data Engine appliances must also be converted to OL8. After converting to OL8, the appliances will be running MariaDB version 10.4.31. See *Converting to OL8*.

- All Database Server, Administration Portal, and/or Data Engine appliances must be upgraded to AP2 version 8.14.26 (Gelato), with the **libem7 IPC** service running. See *Step 3*, below.

- If you are currently deploying SL1 on AWS using Aurora 2 RDS (MySQL 5.7), you must upgrade to Aurora 3 RDS (MySQL 8.0). You must upgrade your Data Engine to AP2 version 8.14.26 (Gelato) before upgrading to Aurora 3. For assistance, contact ScienceLogic Support.

- The SL1 system must be healthy. Review *SL1 Self-Monitoring* to ensure that there are no issues with the system. Specifically, make sure that there are no rows behind, because any rows behind will be lost when the collector is converted. Also, ensure that there is no excessive CPU or swap usage, which can have a negative impact on the conversion.

- To prevent non-recoverable conversion failures, ensure that you do not have any ISOs attached and connected to your collector VMs (and no DVD in the drive) before initiating the conversion. For more information, including steps to take to make sure that you do not have any ISOs attached, see the following KB article: https://support.sciencelogic.com/s/article/15253.

- The collectors must be healthy and sized according to the System Requirements sizing guidelines. The collectors will perform detailed pre-checks to ensure they are eligible for conversion. For more information, see *Collector Readiness Checks*.

- Review the list of files and data that is preserved and restored after the in-place re-installation. See *Collector System-Specific Preserved Data*.

- The Database Server or Data Engine appliances must have at least 5 GB free in the **/var** directory to allow for staging.

- The collectors must have 5 GB of free or reclaimable space.

- Collectors hosted in an Azure environment are *not* eligible for conversion.

- Collectors running in Hyper-V using Gen 2 (UEFI) are not eligible for conversion.

Step 2: Upgrade SL1 Appliances to Version 12.1.2

Complete the following steps, in order, to convert your SL1 appliances to OL8:

1. Upgrade all SL1 appliances to version 12.1.2. For more information, see *Upgrading SL1 to Version 12.1.2*.

2. Upgrade all SL1 appliances on OL7 to MariaDB 10.4.29. For more information, see *Automatically Upgrading MariaDB with a Script*.

> **NOTE:** If the MariaDB upgrade fails while trying to pull the **epel** repository from any SL1 appliance, you can disable the repository by editing **/etc/yum.repos.d/epel.repo** and setting **enabled=0** in the file. Then run the MariaDB upgrade again.

3. Go to the *Converting to OL8* chapter to convert the Database Server to Oracle Linux 8. As part of the OL8 conversion, the MariaDB version will be upgraded to 10.4.31.

4. If you are currently deploying SL1 on AWS using Aurora 2 RDS (MySQL 5.7), upgrade to Aurora 3 RDS (MySQL 8.0). For assistance, contact ScienceLogic Support. If this does not apply to you, you can skip this step.

Step 3: Upgrade SL1 Appliances to AP2 Version 8.14.26 (Gelato)

AP2 version 8.14.26 (Gelato) includes a new *Convert to OL8* option on the **Collector Groups** page of the SL1 user interface. This new option lets you run the conversion process on one of more of your existing collector groups.

> **IMPORTANT:** AP2 version 8.14.26 (Gelato) requires OL8, so any SL1 appliance that is running the nextui service needs to be updated to OL8 before you install Gelato on that appliance.

For more information, see *Installing or Uninstalling AP2 Gelato* at the SL1 AP2 Release Notes site.

When you have AP2 version 8.14.26 (Gelato) installed, make sure the **libem7 IPC** service running.

Step 4: Use the Command-line Interface to Stage the .iso File

> **IMPORTANT:** A collector must have at least 90 GB of hard disk space and at least 6 GB of RAM before you start the conversion process. If the collector does not have 90 GB or more of space, you should increase the disk size to the 90 GB minimum so you can use the conversion utility, or you will need to *re-ISO the collector* instead.

To convert your collectors to OL8 with the command-line interface:

1. On the active Database Server or Data Engine, enable the **dist-upgrade** manager service by running the following command:

   ```
   sudo systemctl start siloupdate-dist-upgrade.service
   ```

   > **NOTE:** Do not set this service to `enabled`. You should start the service manually on the active Database Server appliance when you are ready to use it.

2. Download the **em7_x86_64_12.1.2-<*version*>.el8.iso** file from the ScienceLogic Support site at https://support.sciencelogic.com/s/release-file/aBtVL0000000KRl0AM/1212.

3. Upload or copy the **.iso** file to a temporary location on the active Database Server or Data Engine. The recommended directories are the **/tmp** or the **/var/tmp** directories, and they should have enough free space. You can verify the free space prior to upload by using `df -h` to make sure that there is at least 5 GB free space in the temporary location to which you are uploading.

4. After the upload is complete, run the following command to check the sha512sum signature of the **.iso** file.

   ```
   sha512sum <full path to ISO>
   ```

   > **IMPORTANT:** It is critical that you make sure the signature of the **.iso** file is the same as the signature listed on the ScienceLogic portal. If the signature is not the same, this indicates corruption in the **.iso** file, which will lead to failures during the conversion procedure.

5. Import the **.iso** file to the local filestore, and make a note of the file ID after the file has been uploaded. The ID is `2521` in the following example:

   ```
   sudo pcli filestore upload -f <full path to ISO>

   Uploading file em7_x86_64_12.1.2-442.el8.iso

   Uploading file : 4099446784 bytes [04:25, 15433673.17 bytes/s]

   File uploaded successfully with ID 2521
   ```

6. After you upload the **.iso** file to the filestore, remove the file from the temporary location to free up disk space:

   ```
   sudo rm <full path to ISO>
   ```

7. Create a distribution image policy referencing the image you just imported:

   ```
   sudo siloupdate dist-upgrade-manager set-image <file ID from step 4>
   ```

8. Create a staging policy to stage the image to all valid collectors:

```
sudo siloupdate dist-upgrade-manager enable-staging
```

The distribution upgrade manager will cache the **.iso** file and the upgrade script to the local system and begin to stage it across all OL7 collectors. This process can take several minutes to stage out the script and the **.iso** file to the local filesystem.

9. Go to *Step 5* to complete the collector conversion.

Step 5: Use the Collector Groups Page in SL1 to Convert Collectors to OL8

You can manage the collector conversion process on the **Collector Groups** page (Manage > Collector Groups) in SL1.

For each collector group, this page displays one of the following labels in the *Status* column:

- **OL8 Staging Required**. Staging of the OL8 **.iso** file is required for at least one collector in this group. It is possible that distribution upgrade script are in the process of being sent out to the collectors. To stage your collector groups to prepare them for the conversion process, see *Step 4*, above.

- **File Verification in Progress**. The OL8 **.iso** file is currently being validated on at least one Collector prior to progressing to the next step. This status will be displayed at multiple points through the conversion process as the file has been transferred to collectors, staged, and conversion.

- **OL8 Staging in Progress**. The **.iso** file and the distribution upgrade script have been fully sent to the collector. The collector is staging them locally and preparing for the in-place upgrade. During this stage, the collector readiness checks are run.

- **OL8 Staging Failed**. Something failed with staging on at least one collector in a collector group, typically a collector readiness check. The system will automatically retry staging and the status will go back to *OL8 Staging Required*. The system will validate that the **.iso** file and script were transferred correctly and fix any corruption during transfer. Then the system will retry the staging again.

- **Ready for OL8 Conversion**. All readiness checks have passed, staging is complete, and you can convert the OL7 collectors to OL8.

- **OL8 Conversion in Progress**. One or more collectors in a collector group has had the conversion process triggered on it.

- **Collectors Partially Converted**. The conversion has completed for at least one collector in the collector group, other collectors within the collector group have not started conversion and no failures have occurred.

- **OL8 Conversion Failed**. The conversion has failed on at least one collector in this collector group; staging was complete for all collectors within the collector group.

- **OL8 Conversion Complete**. The OL8 conversion is complete for all collectors in the collector group.

> **NOTE:** Until the **.iso** file is cached on the local filesystem and begins to stage to your collectors, the *Status* column will be blank. This process can take several minutes to cache.

The following image shows two collector groups that are fully staged and ready for OL8 conversion:

The following image shows two collectors groups, the first in progress with staging, and the second in need of staging.

NOTE: Staging takes place in the background, and this process can take several minutes to several hours, depending on the speed of the connection between the collectors.

IMPORTANT: In the following procedure, you have the ability to trigger the conversion of *all* collectors within a collector group if all of the collectors have a status of *Ready for OL8 Conversion*. ScienceLogic does not recommend this, because all collectors in your group will be taken offline at roughly the same time. It can take several minutes for the conversion to happen, and collectors will be unavailable while the conversion is in progress. If any issue occurs, the collectors might not come back: because collectors in a group are usually identical, all collectors will likely be impacted by the issue, resulting in an extended outage for the collector group.

To complete the OL8 conversion for your collectors:

1. In SL1, go to the **Collector Groups** page (Manage > Collector Groups) .

2. If you need to stage your collector groups, a banner at the top of the **Collector Groups** page displays, and the **Status** for each unstaged collector group displays *OL8 Staging Required*. To stage your collector groups to prepare them for the conversion process, see *Step 4*, above.
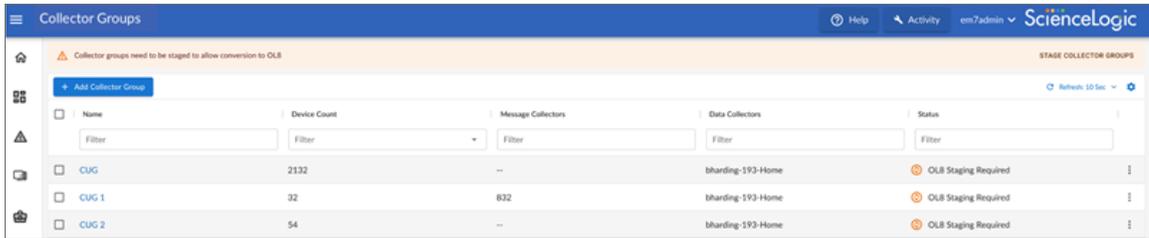
3. When the **Status** for a collector group displays *Ready for OL8 Conversion*, click the Actions icon (⋮) for a collector group and select *Convert to OL8*. The **Select Collectors to Convert to OL8** modal displays all collectors for that collector group.
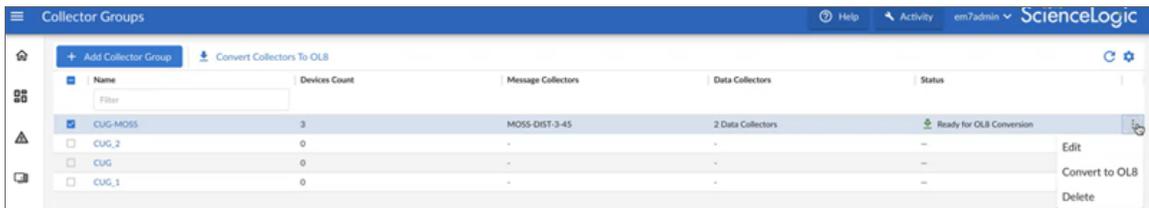
> **TIP:** If you want to be convert multiple collector groups listed on the **Collector Groups** page, click the check boxes for one or more collector groups and click the **[Convert Collectors To OL8]** button. The **Select Collectors to Convert to OL8** modal displays all of the collectors for the selected collector groups.

4. On the **Select Collectors to Convert to OL8** modal, verify that the *OS Version* is *OL7* and the *Conversion Status* is *Ready for OL8 Conversion* for your collectors. If needed, click the *Collector Name* and review the details for that collector.
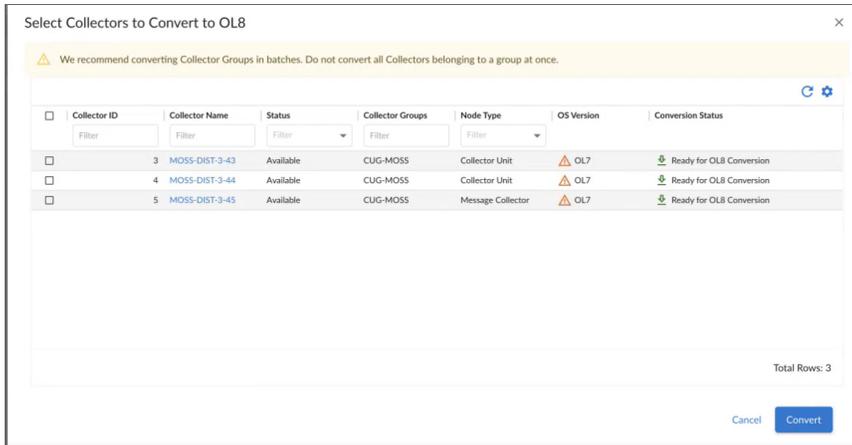
> **TIP:** If your collectors or collector groups are toggling between *OL8 Staging Failed*, *OL8 Staging Required*, and *OL8 Staging in Progress*, most likely the collector readiness checks are failing. You will need to troubleshoot why they are failing and either correct the issue or re-ISO manually. For more information, see *Troubleshooting the Collector Conversion Process*.

5. Click the check boxes to select one or more collectors to convert to OL8. When converting collectors, you can select all but one data collector in a specific collector group.

    If a collector group has failover enabled, make sure that the remaining number of available data collectors within a collector group is equal to or greater than the **Collectors Available for Failover** value on the **Collector Group Management** page in SL1. This will minimize impact to collection within that collector group, and enable collector failovers to function properly.

    Similarly, if the collectors are production collectors, make sure to account for collector outage and rebalance by selecting just a few collectors from each group. You must ensure your collector groups are sized correctly for failover and rebalance.

6. Click **[Convert]** to start the conversion process.

   During this process, the collectors will go offline and perform an in-place re-installation to OL8. Because this is an offline process, there is no visibility in to what is happening from the user interface, and the status may toggle while the conversion is in progress. The only way to see the progress or errors on the collector is to view the console output. For more information, see *Troubleshooting the Collector Conversion Process*.

   > **NOTE:** Collectors running in AWS will be in a *Stopped* state after the in-place reinstallation. You will need to start these systems, and they will be running OL8.

   After the collectors have been converted to OL8, the **OS Version** is updated to *OL8*, and MariaDB is upgraded to 10.4.31 on your Message Collectors and Data Collectors.

7. Repeat steps 2-6 for your remaining collectors.

You can run the following command on a converted collector to make sure that the collector was successfully migrated to OL8:

```
cat /etc/os-release
```

Review the feedback and look for `Oracle Linux 8`:

```
[em7admin@cu-10-2-23-66 ~]$ cat /etc/os-release
NAME="Oracle Linux Server"
VERSION="8.10"
ID="ol"
ID_LIKE="fedora"
VARIANT="Server"
VARIANT_ID="server"
VERSION_ID="8.10"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Oracle Linux Server 8.10"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:oracle:linux:8:10:server"
HOME_URL="https://linux.oracle.com/"
BUG_REPORT_URL="https://github.com/oracle/oracle-linux"

ORACLE_BUGZILLA_PRODUCT="Oracle Linux 8"
ORACLE_BUGZILLA_PRODUCT_VERSION=8.10
ORACLE_SUPPORT_PRODUCT="Oracle Linux"
ORACLE_SUPPORT_PRODUCT_VERSION=8.10
```

> **IMPORTANT:** On collectors that were deployed using an ISO file for SL1 11.2.0 or older, native SNMP Dynamic Applications on the devices monitored on the newly converted collector stop collecting data after the collectors were converted to OL8. For more information and a workaround for this issue, see https://support.sciencelogic.com/s/article/16517.

Troubleshooting the Collector Conversion Process

If the *Conversion Status* for a has been displaying the same status for a long period of time, and you need more information about what is happening, refer to the following sections to help determine what is happening in the conversion process.

Check, start, or restart the distribution upgrade service

Run the following command on the Primary/Active Database Server/Data Engine to check if the service is running:

```
sudo systemctl status siloupdate-dist-upgrade.service
```

Review the output to see if the service is `active (running)`:

- `siloupdate-dist-upgrade.service - SL1 Distribution Upgrade Manager`
   `Loaded: loaded (/usr/lib/systemd/system/siloupdate-dist-upgrade.service; enabled; vendor preset: disabled)`
  `Drop-In: /etc/systemd/system/siloupdate-dist-upgrade.service.d`
          `└─db-diff.conf`
   `Active: `**`active (running)`**` since Mon 2024-07-15 18:05:33 UTC; 1h 59min ago`
  `Process: 2191146 ExecStartPre=/bin/sh -c sed "s/CREATE TABLE/CREATE TABLE`
  `IF NOT EXISTS/" /usr/share/siloupdate/dist-upgrade/schema/em7_db/*/*.sql |`
  `/opt/em7/bin/mysql master_platform>`
 `Main PID: 2191155 (python3.6)`
    `Tasks: 20 (limit: 199874)`
   `Memory: 55.0M`
   `CGroup: /system.slice/siloupdate-dist-upgrade.service`
          `└─2191155 /usr/bin/python3.6 -m siloupdate.dist_upgrade -v`

To start the distribution upgrade service, run the following command on the Primary/Active Database Server/Data Engine:

```
sudo systemctl start siloupdate-dist-upgrade.service
```

To restart the distribution upgrade service, run the following command on the Primary/Active Database Server/Data Engine:

```
sudo systemctl restart siloupdate-dist-upgrade.service
```

> **NOTE:** Do not set the distribution upgrade service to `enabled`.

Check the logs for the distribution upgrade service (Database Server/Data Engine)

Run the following command and review the output:

```
sudo journalctl -f -u siloupdate-dist-upgrade.service
```

This log will contain information about what the service is working on.

The following messages indicates that a collector readiness check has failed:

```
Jul 01 20:31:54 test-db1 siloupdate[2689424]: Finished System Checks
```

```
Jul 01 20:31:54 test-db1 siloupdate[2689424]: Error: Script exited due to
an error condition.
```

If you look higher in the log, you will see messages like the following, which indicate the tests that are being run and failing:

```
Jul 01 20:31:00 test-db1 siloupdate[2689424]: [DEBUG] [mid=5] running
('/bin/sudo', 'siloupdate', 'dist-upgrade', '-s', '-x',
'3bbf31caffc70df60d8f99486964a7f95bc4e84d7cd65b8b39bf33562e59dc540516b13e
66c27c935cb96aa29731fb96996d11f765f2f7dc608c42a29cb0d32c')
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: [DEBUG] [mid=5] dist-upgrade
-s stdout:
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: Running System Checks
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - SL1 Version
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - OS Version
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - Appliance licensed
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - Appliance model type
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: FAIL - Appliance RAM size
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - Network configuration
```

```
Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - Appliance Primary IP
address

Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - Appliance Default
Gateway

Jul 01 20:31:39 test-db1 siloupdate[2689424]: PASS - Appliance DNS

Jul 01 20:31:49 test-db1 siloupdate[2689424]: [DEBUG] WATCHDOG=1

Jul 01 20:31:54 test-db1 siloupdate[2689424]: [DEBUG] [mid=5] dist-upgrade
-s stdout:

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Phomehone information

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - ISO for reimaging in
local filestore

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Determine space
required for reimaging

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Volume group
validation

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Reimage volume space
requirements

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Account validation
for root

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Account validation
for em7admin

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Account validation
for sl1user

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Determine Boot
identifiers

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Determine physical
disk layout

Jul 01 20:31:54 test-db1 siloupdate[2689424]: PASS - Determine physical
disk identifiers
```

```
Jul 01 20:31:54 test-db1 siloupdate[2689424]: Finished System Checks

Jul 01 20:31:54 test-db1 siloupdate[2689424]: Error: Script exited due to
an error condition.

Jul 01 20:31:54 test-db1 siloupdate[2689424]: [DEBUG] [mid=5] dist-upgrade
-s status code: 1

Jul 01 20:31:54 test-db1 siloupdate[2689424]: [DEBUG] [mid=5] siloupdate
returncode: 1

Jul 01 20:31:54 test-db1 siloupdate[2689424]: [DEBUG] [mid=5] remote
staging failed
```

From this log section, you can see that it was operating on module is 5 (`mid=5`). Looking at the messages, it failed on the RAM size of the appliance. We can also see that the `returncode` was `1`. Any return code that is not a `0` is a failure.

Messages like the following in the log indicate the process is running, but there is no active work for it at the moment:

```
Jul 01 06:53:44 test-db1 siloupdate[1658074]: [DEBUG] WATCHDOG=1

Jul 01 06:56:14 test-db1 siloupdate[1658074]: [DEBUG] WATCHDOG=1

Jul 01 06:58:44 test-db1 siloupdate[1658074]: [DEBUG] WATCHDOG=1

Jul 01 07:01:14 test-db1 siloupdate[1658074]: [DEBUG] WATCHDOG=1

Jul 01 07:03:44 test-db1 siloupdate[1658074]: [DEBUG] WATCHDOG=1
```

The following messages in the log indicate that the connection to the collector was lost or is unstable. Please make sure you have stable and reliable network connectivity and restart the **dist-upgrade** service:

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: [ERROR]
Task exception was never retrieved

Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: future:
<Task finished coro=<_SingleCoroutineMixin._try_coro() done, defined at
/usr/lib/python3.6/site-packages/siloupda>

Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: Traceback
(most recent call last):

Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 149, in _try_coro
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 401, in stage
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 382, in _check_platform
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 259, in do
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 236, in _await_module_cmd
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 254, in do_live
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 208, in module_cmd_live
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/usr/lib/python3.6/site-packages/siloupdate/dist_upgrade/manager.py",
line 190, in _create_spool
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/opt/em7/lib/python3/aiohttp/client.py", line 690, in __aenter__
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: self._resp
= yield from self._coro
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/opt/em7/lib/python3/aiohttp/client.py", line 277, in _request
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: yield from
resp.start(conn, read_until_eof)
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/opt/em7/lib/python3/aiohttp/client_reqrep.py", line 624, in start
```

```
Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: (message,
payload) = yield from self._protocol.read()

Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: File
"/opt/em7/lib/python3/aiohttp/streams.py", line 554, in read

Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]: yield from
self._waiter

Jul 02 16:20:42 ip-10-152-0-139.ec2.internal siloupdate[26224]:
aiohttp.client_exceptions.ServerDisconnectedError: None
```

Check the distribution upgrade logs on collectors

When staging to a collector, a script runs to determine if the collector is eligible for the distribution upgrade and to complete the staging process. When this script runs, the output is sent back to the Database Server or Data Engine and logged by the distribution upgrade service. There might be times when it is easier to look at the log from the collector directly or to retrieve it so it can be reviewed by ScienceLogic Support. This log can be viewed in two ways, which are covered below.

Option 1: View the log from the Database Server or Data Engine

Each collector stores the logs from the script run in its local database. From the Database Server or Data Engine you can connect to the collector and retrieve the last run by running a query like the following:

```
silo_mysql -h 127.0.0.3 -P7707 -e "SELECT log_file FROM
sysinfo.laboratory_logs ORDER BY run_date DESC LIMIT 1"
```

> NOTE: You will need to adjust the IP address to reflect the collector you want to query and possibly add the flags for specifying the password.

Option 2: View the log from the collector

You can log in the collector and view the log directly. The log from the current run is in **/var/log/em7/recovery.log**. The log from the previous run is **/var/log/em7/recovery.log.bz2**.

When viewing logs on the collectors, you can run the following command to find just the lines indicating a check failed:

```
sudo grep FAIL /var/log/em7/recovery.log
```

Check if files are cached for staging

Run the following command to check if the files are cached:

```
ls -lh /var/cache/siloupdate/dist-upgrade/
```

> **NOTE:** Sizes may differ on your system.

You should see output similar to the following if the files are fully cached:

```
total 3.9G

-rw-r--r-- 1 s-em7-core s-em7-core 55K Jun 27 23:17 sl1-dist-upgrade

-rw-r--r-- 1 s-em7-core s-em7-core 3.9G Jun 27 23:21 sl1-image.iso
```

Check that an image policy exists

The image policy should be automatically created when following the steps in the *initial import and setup*.

The following query will list all image policies:

```
silo_mysql -e "SELECT * FROM master_platform.image_policy"
```

You should expect a result similar to the following:

```
+-----------------+---------------+-----------------------+---------------
-+-----------------------+
| image_policy_id | image_file_id | image_remote_filename | script_file_id
| script_remote_filename |
+-----------------+---------------+-----------------------+---------------
-+-----------------------+
|               1 |          2520 | sl1-image.iso         |          2521
| sl1-dist-upgrade      |
+-----------------+---------------+-----------------------+---------------
-+-----------------------+
```

The file id numbers might be different for your setup. The `image_file_id` should match the file id number of the ISO you imported in to the filestore. The `script_file_id` will be populated automatically. Typically you will have only one image policy on a system, but you could have more in some situations.

If there is no image policy, see the *Initial Setup and Staging section* to create it.

Check that an image staging policy exists

The image staging policy should be automatically created when following the steps in the initial import and setup.

The following query will list all image staging policies.

```
silo_mysql -e "SELECT * FROM master_platform.image_staging_policy"
```

You should expect a result similar to the following:

```
+------------------+----------------+---------+
| staging_policy_id | image_policy_id | enabled |
+------------------+----------------+---------+
|                1 |              1 |       1 |
+------------------+----------------+---------+
```

Typically you will only have one image staging policy, but in some situations you could have more. The `image_policy_id` should reference the image policy. When `enabled` is `1`, the policy is enabled for staging, and when it is `0` it is disabled.

If there is no image staging policy, see the *Initial Setup and Staging section* to enable staging.

Checking appliance image status

Run the following command to check the appliance image status:

```
silo_mysql -e "SELECT mid, state, status FROM master_platform.appliance_
image_status"
```

You should expect a result similar to the following:

```
+-----+----------+------------------------------------------------+
| mid | state    | status                                         |
+-----+----------+------------------------------------------------+
|   8 | transfer | {"script_ready": true, "db_space": 77577342976} |
|   9 | transfer | {"script_ready": true, "db_space": 77585526784} |
|  10 | transfer | {"script_ready": true, "db_space": 77595070464} |
|  11 | transfer | {"script_ready": true, "db_space": 77574176768} |
|  12 | transfer | {"script_ready": true, "db_space": 77574107136} |
+-----+----------+------------------------------------------------+
```

You can see that the system has a `state` of `transfer`, which indicates that the image is being staged to the collectors.

Checking appliance image staging

Run the following command to check the staging progress of the images sent to each collector:

```
silo_mysql -e "SELECT mid, (bytes_sent / bytes_total) * 100 AS pct,
remote_filename FROM master_platform.appliance_image_staging"
```

You should expect a result similar to the following:

```
+-----+----------+-----------------+
| mid | pct      | remote_filename |
+-----+----------+-----------------+
|   8 | 100.0000 | sl1-dist-upgrade |
```

```
|    8 |   79.3187 | sl1-image.iso     |
|    9 |  100.0000 | sl1-dist-upgrade |
|    9 |   79.6512 | sl1-image.iso     |
|   10 |  100.0000 | sl1-dist-upgrade |
|   10 |   79.6768 | sl1-image.iso     |
|   11 |  100.0000 | sl1-dist-upgrade |
|   11 |   79.6512 | sl1-image.iso     |
|   12 |  100.0000 | sl1-dist-upgrade |
|   12 |   79.6768 | sl1-image.iso     |
+-----+----------+------------------+
```

You can see that the percent (`pct`) complete for the script and ISO being staged to each collector.

If you find that a file is not being staged (`pct` is stuck at the same value for a long period of time), the connection might have been lost. Check the logs for the distribution upgrade service. If no active work is happening and the percent complete is still not incrementing, restart the distribution upgrade service by running the following command on the Primary/Active Database Server/Data Engine:

```
sudo systemctl restart siloupdate-dist-upgrade.service
```

Force restage images to collectors

After staging is marked as complete, the system will no longer check the appliances. There are some circumstances where manual changes might have been made to an appliance to invalidate the staging. To force an appliance to restage, run the following query:

```
silo_mysql -e "DELETE FROM master_platform.appliance_image_status WHERE
mid = <module id>"
```

This command removes the entry indicating that the appliance is fully staged. Within a few minutes, the staging process should start up and restage the appliance.

Collector has been converted to OL8 but the platform still lists OL7

After you convert a collector to OL8, if SL1 still indicates that it needs conversion and the **Platform** is still listed as *el7* on the **Appliances** page (System > Settings > Appliances), you will need to check to ensure that the collector is still reachable by the Database Server or the Data Engine. If you are using Phone Home, make sure that the Phone Home connection is active.

You should check the **Collector Status** page (System > Monitor > Collector Status) and make sure the **Collector State** is *Available*. If the collector is marked as *Available*, you will need to manually refresh the appliance status with the following command:

```
sudo siloupdate refresh-appliance -m <module id>
```

This command will take a few moments to run, and when it is finished, the **Platform** should be marked as *el8* on the **Appliances** page, and the conversion will be marked as a success.

Collector MariaDB version is correct but highlighted

After upgrading all of your SL1 appliances to 12.1.2 and converting all appliances to Oracle Linux 8, a known issue might cause the **MariaDB** field for your appliances to be highlighted on the **Appliance Manager** page (System > Settings > Appliances) in SL1. If the appliances are showing the correct MariaDB version of 10.4.31, this highlighting can be safely ignored.

Collector never displays a status of "Ready for OL8 Conversion"

The collector not displaying a status *Ready for OL8 Conversion* can occur for a number of reasons. In most cases, it is caused by the collector failing one of the readiness checks.

To determine the cause of the issue:

1. Check the logs of the distribution upgrade service for any issues with the service or issues staging the collectors.

2. Review the following sections to ensure the policies are created and the staging status:
    *   Check if files are cached for staging .

    *   Check that an image policy exists.

    *   Check that a image staging policy exists.

    *   Check appliance image staging.

3. Check the distribution upgrade logs on the collectors.

If checking these options do not resolve the issue, or if you are not sure how to proceed, contact ScienceLogic Support.

Collector fails to return to service after initiating the conversion

The expected time range for a collector to reinstall and return to service is 10 to 15 minutes. This time frame is a general guide based on ScienceLogic testing. Various factors can influence the time it actually takes. If it takes longer than 15 minutes, you should investigate to determine what has happened. Being able to access the console to view the output is required in order to determine the state of the collector after reinstallation has begun.

Examine the output to determine if the system is in the process of installation. If it is, wait for it to complete.

If the system is hung, capture a screenshot for review by ScienceLogic Support. There will be no way for Support to fix a system in this state. You will need to reboot. Depending on where the collector hung during the installation, you might need to re-ISO the collector instead. The screenshot you gather will be used to determine if future changes can be made to address the problem.

Collector on AWS stops in the middle of conversion

When converting AWS SL1 collectors to Oracle Linux 8 (OL8), you might experience an issue where the collectors stop in the middle of the conversion and do not restart automatically, leaving the conversion process hanging. If this occurs, you can go into the AWS console and restart the collector manually.

Collector Conversion Informational Details

Converting a collector from OL7 to OL8 at a high level involves the following processes:

1. Send an ISO and script to the collector that will be used for an in-place re-installation.

2. Perform system checks to determine if the collector group or Message Collector is eligible for in-place re-installation.

3. Create a recovery volume in the unused space.

4. Stage the ISO and preserve system specific data.

5. Build a custom kickstart file to use for in-place reinstallation.

6. Use kexec to activate the kernel for the in-place reinstall process.

Collector Readiness Checks

The following system checks are done on the collector to ensure it is eligible for in-place reinstallation.

- SL1 Version. Verifies that this is a 12.1.x version.

- Appliance licensed. Verifies that system is licensed. This test should always pass, as collectors automatically get licensed. If this check fails, there are probably deeper problems on this system.

- Appliance model type. Verifies that this system is an Data Collector or Message Collector.

- Appliance RAM size. Verifies that this collector meets the minimum RAM requirements for a collector of 6 GB.

- Appliance Disk Size. Verifies that this collector meets the minimum disk requirements for a collector of 90 GB. This check verifies that the Volume Group is correctly sized for a 90 GB disk.

- Network configuration. Checks for a primary interface on this system. A primary interface is the one that has the default gateway. Also checks that the interface is set to either `none` or `dhcp`.

- Appliance Primary IP address. Checks for an IP on the interface identified as primary.

- Appliance Default Gateway. Checks for the default gateway address.

- Appliance DNS. Checks to see if SL1 can extract valid DNS entries on the system.

- Appliance running in Azure. Azure is not supported for in-place updates due to issues with Azure.

- Appliance is Gen 2 Hyper-V VM. Generation 2 Hyper-V VMs are not supported for in-place updates due to issues with Hyper-V.

- Phone Home information. Checks if Phone Home is in use and if the required files are accessible.

- ISO for reimaging in local filestore. Checks if the ISO has already been uploaded.

- Determine space required for reimaging. Checks if SL1 can calculate the required space needed to do the reimaging process.

- Sufficient scratch space available. This check is currently skipped when using the ISO process (this check is intended to be used if SL1 ever supports an online process).

- Volume group validation. Checks for the em7vg volume group.

- Reimage volume unmounted. If there is already a reimage volume, checks that it is unmounted. Script will attempt to unmount and proceed.

- Reimage volume space requirements (with swap reclaim). Checks that the reimage volume is the correct size. If the volume does not exist, checks that there is enough space to create it. If not enough space, checks swap size and determines if reclaiming swap will allow it to be created.

- Account validation for root. Checks for the account and extracts the password hashes.

- Account validation for em7admin. Checks for the account and extracts the password hashes.

- Account validation for sl1user. Checks for the account and extracts the password hashes.

- Determine EFI identifiers. If system is EFI, boot check to identify the partition.

- Determine boot identifiers. Checks for the boot partition.

- Determine physical disk layout. Checks that there is only one physical volume as part of the volume group.

- Determine physical disk identifiers. Checks that SL1 can identify the physical disk.

Collector System-Specific Preserved Data

The following files and data are preserved and restored after the in-place reinstallation.

Files:

- /etc/silo.conf
- /etc/resolv.conf
- /etc/dnsmasq-resolv.conf
- /etc/chrony.d/*
- /opt/em7/share/config/firewalld.d/80_allow_db_conns.conf
- /etc/siteconfig/firewalld-rich-rules.siteconfig
- /home/em7admin/.ssh/*
- /etc/rsyslog.d/siteconfig.d/*
- /etc/krb5.conf
- /etc/phonehome

Data:

- Primary IP address and default gateway (secondary IPs and static routes are not preserved)
- Phone Home connection information
- em7admin password
- sl1user password
- root password

Option 2: Redeploying Existing Collectors with New Collectors (Re-ISO)

> **IMPORTANT:** Before converting the collectors to Oracle Linux 8, you will first need to upgrade your Database Server to SL1 version 12.1.2. For more information, see *Upgrading SL1 to Version 12.1.2*.

If you consider the collector appliance as a "throwaway" appliance, or if you have very few collectors deployed, then the best option is to *replace* (re-ISO) the existing collectors by deploying new 12.1.x OL8 collectors instead of keeping the original collectors.

Complete the following steps:

1. *Step 1: Prepare for the Re-ISO*

2. *Step 2: Delete the Collector*

3. *Step 3: Install and Register the New Collector* (if you are replacing the collector)

> **IMPORTANT:** ScienceLogic recommends that you read through this entire chapter before starting the conversion. Contact your ScienceLogic representative for assistance or more detailed guidance, if needed.

Step 1: Prepare for the Re-ISO

1. Make sure the ISO image is downloaded. If your SL1 stack is on a version that you can only do an update, the recommended plan is to install the collector on the closest lower version.

2. Decide if you will be reinstalling the new collector "in place" (replacing the existing collector) or if you plan to install the new collector "alongside" the old one and then switch over to the new collector.

   ○ If you are replacing the collector, it is easier to simply delete the collector from SL1 completely before reinstalling. This requires removing the collector from the collector group it belongs to before deleting it. Then you can run the corresponding procedure for reinstalling the collector itself.

   ○ If you are installing alongside the old collector, it will be easier to treat it as a "net-new" collector, meaning that the new collector should get its own unique resources, such as IP addresses.

Step 2: Delete the Collector

> **NOTE:** It is out of the scope of this topic to cover the steps for decommissioning old collectors, standing up new ones within virtualization platforms, or standing up physical hardware.

Regardless of whether the installation is alongside or a wholesale replacement of the collector, you should following the installation process at *Installing and Configuring an SL1 Collector* in the ScienceLogic Product Documentation.

- **If you are installing alongside the old collector**, complete the *Installing and Configuring* procedures above before continuing. Follow the steps to connect the collector to the SL1 stack before continuing, including registering it to SL1 and adding it to the correct SL1 collector group.

- **If you are replacing the collector**, the following steps will stop any collections that still work for devices aligned to the existing collector. Plan according to your organization's change management procedures.

If you have multiple collectors in the SL1 collector group:

1. In SL1, go to the **Collector Groups** page (System > Settings > Collector Groups.

2. Locate the collector group to which the collector belongs and click its edit icon ( ).

3. In the fields at the top of the page, unselect the collector in question and click **[Save]**.

4. Go to the **Appliances** page (System > Settings > Appliances).

5. Locate the collector appliance and click the delete icon ( ) on the right.

6. Click **[OK]** to delete the collector.

7. Follow the proper procedure for decommissioning the collector appliance itself.

8. **If you are installing alongside the old collector**, go to the **Collector Groups** page (System > Settings > Collector Groups, locate the collector group, and click the lightning bolt icon ( ) on the right side to rebalance the devices in this collector group across the new list of collectors. You do **not** need to continue to Step 3.

9. After the collector is rebuilt, add the collector to the **Appliances** page (System > Settings > Appliances). Make sure to specify the DB User and DB Password values when adding the collector. See the *Caveats*, below.

10. Add the new collector to the old collector group on the **Collector Groups** page (System > Settings > Collector Groups by locating the collector group to which the collector belongs and clicking its edit icon ( ). Press the **[Ctrl]** key and select the rebuilt collector. You will see all collectors highlighted in the *Collector Selection* field. Click **[Save]** to save the configuration.

11. If collector failover is not enabled on the collector group, you can click the lightning bolt icon ( ) to rebalance devices in the collector group across all available collectors that belong to that collector group.

If you have a collector group with only a single collector:

1. Go to the **Collector Groups** page (System > Settings > Collector Groups and create a virtual collector group. A *virtual collector group* is a collector group without any aligned collectors.

2. Go to the **Device Manager** page (Devices > Device Manager). Filter all the devices that belong to the collector group that the rebuilt collector is aligned with. Select those devices and move them into the virtual collector group you created in Step 1, above. This is done to allow deletion of the old collector and collector group from the system.

3. Go to the **Collector Groups** page (System > Settings > Collector Groups and note that a delete icon ( ) is now active on the old collector group. Click the delete icon ( ) to remove the old collector group.

4. Go to the **Appliances** page (System > Settings > Appliances), locate the collector, and click the delete icon ( ) on the right.

5. Click **[OK]** to delete the collector.

6. Follow the proper procedure for decommissioning the collector appliance itself.

7. After the collector is rebuilt, add the collector to the **Appliances** page (System > Settings > Appliances). Make sure to specify the DB User and DB Password values when adding the collector. See the *Caveats*, below.

8. Recreate the old collector group with the rebuilt collector aligned.

9. Go to the **Device Manager** page (Devices > Device Manager), select all the devices that you temporarily moved to the virtual collector unit, and move them back to the newly added collector group.

Step 3: Install and Register the New Collector (if you are Replacing the Collector)

1. Follow the steps in *Installing and Configuring an SL1 Collector* in the ScienceLogic Product Documentation to get the new collector installed and registered to SL1.

2. Decommission the previous collector by first removing it from its collector group, and then remove its registration from SL1.

Caveats

Consider the following issues when planning and troubleshooting your collector updates:

- One issue that many users encounter is assuming that a collector with the same IP address can connect back to SL1. However, SL1 checks for unique information and will be able to tell if the collector attempting to connect to it from an IP Address is not the same collector that it used to be. As a result, if you are reusing the IP Address for a collector, you should remove that registered collector before attempting to connect the new collector.

- If you used the SL1 11.3.0 ISO, the DB User should be "clientdbuser", and the DB Password is whatever you selected during the installation for the backend access.

- If you used the SL1 11.2.0 or older ISOs, then the DB User is "root" and the DB Password is default password. To change the default MariaDB password on the collector, see *Changing Passwords* in the ScienceLogic Product Documentation.

Chapter

3

Step 3: After You Convert to OL8

Overview

This chapter provides an overview of post-conversion activities for an Oracle Linux 8 (OL8) conversion.

> **CAUTION:** To avoid outages or down time, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL8 before attempting to convert a production system.

This chapter covers the following topics:

Validate the Conversion

After completing the conversion to Oracle Linux 8, ScienceLogic recommends that you run the following validations:

1. Run `sudo licensed_state` to check if the system is licensed. If it is not, you must license the system. For more information, see *Licensing and Configuring an Appliance* in the SL1 Product Documentation.

2. Run `system_status.sh` and review the output for problems.

3. Run `/opt/em7/backend/collector_check.py` to check collector availability.

4. Log in to the SL1 user interface and make sure that the system is working as expected.

5. In SL1, go to the **Appliances** page (System > Settings > Appliances) and review the **Platform** column. If an appliance has *el8* in that column, then the appliance is running OL8, while *el7* means the appliance is running OL7. You can also check the **MariaDB** column for the proper version of the database.

6. Another way to verify that the Database Servers or the Data Engine are successfully migrated to OL8 is to use SSH to access the server and run the following command:

   ```
   cat /etc/os-release
   ```

   Review the feedback and look for `Oracle Linux 8`:

   ```
   [em7admin@ip-10-152-2-186 ~]$ cat /etc/os-release
   NAME="Oracle Linux Server"
   VERSION="8.10"
   ID="ol"
   ID_LIKE="fedora"
   VARIANT="Server"
   VARIANT_ID="server"
   VERSION_ID="8.10"
   PLATFORM_ID="platform:el8"
   PRETTY_NAME="Oracle Linux Server 8.10"
   ANSI_COLOR="0;31"
   CPE_NAME="cpe:/o:oracle:linux:8:10:server"
   HOME_URL="https://linux.oracle.com/"
   BUG_REPORT_URL="https://github.com/oracle/oracle-linux"

   ORACLE_BUGZILLA_PRODUCT="Oracle Linux 8"
   ORACLE_BUGZILLA_PRODUCT_VERSION=8.10
   ORACLE_SUPPORT_PRODUCT="Oracle Linux"
   ORACLE_SUPPORT_PRODUCT_VERSION=8.10
   ```

Post-conversion Tasks

For all conversion procedures, the Destination system will be a singular system when you complete the conversion:

- An All-in-One Destination system will be a All-in-One system.
- A standalone Database Server, a High Availability (HA), a Disaster Recovery (DR), or a combined HA and DR (HA+DR) system will all be a standalone Database Server.

HA/DR Settings

After you complete the conversion to OL8 and you have verified that the SL1 is working properly, you can implement HA, DR, or HA+DR by following the procedures in the *High Availability & Disaster Recovery Configuration* section of the SL1 Product Documentation.

MariaDB Updates

After the conversion to OL8, all OL8 appliances will be running MariaDB 10.4.31. Until you run the *collector conversion*, your OL7 collectors should be running MariaDB version 10.4.29.

NOTE: If the MariaDB upgrade fails while trying to pull the **epel** repository from any SL1 appliance, you can disable the repository by editing **/etc/yum.repos.d/epel.repo** and setting **enabled=0** in the file. Then run the MariaDB upgrade again.

SL1 Email Settings

To make sure that inbound emails continue to be processed after the conversion to OL8, perform the following steps on the Database Server:

1. In the SL1 user interface, go to the **Email Settings** page (System > Settings > Email).
2. Click **[Save]**.

SL1 Appliances Page

The following image is from an **Appliances** page (System > Settings > Appliances) for an SL1 system where all of the appliances have been converted to OL8:

1.	crucible-g-cdb-172	10.64.164.172	Database	n/a		Database: 10.64.164.172	12.1.2 r465	10.4.31	el8	99,000	n/a	1	Yes	--	No
2.	crucible-giga-ap-183	10.64.164.183	Administration Portal	n/a			12.1.2 r465		el8	n/a	n/a	54	Yes	--	No
3.	crucible-giga-cu-158	10.64.164.158	Data Collection Unit	CUG1			12.1.2 r465	10.4.31	el8	n/a	2941	35	Yes	--	No
4.	crucible-giga-cu-159	10.64.164.159	Data Collection Unit	CUG2			12.1.2 r465	10.4.31	el8	n/a	2941	36	Yes	--	No
5.	crucible-giga-cu-173	10.64.164.173	Data Collection Unit	CUG3			12.1.2 r465	10.4.31	el8	n/a	2941	37	Yes	--	No
6.	crucible-giga-cu-174	10.64.164.174	Data Collection Unit	CUG4			12.1.2 r465	10.4.31	el8	n/a	2942	38	Yes	--	No
7.	crucible-giga-cu-175	10.64.164.175	Data Collection Unit	CUG5			12.1.2 r465	10.4.31	el8	n/a	2941	39	Yes	--	No
8.	crucible-giga-cu-176	10.64.164.176	Data Collection Unit	CUG6			12.1.2 r465	10.4.31	el8	n/a	2941	40	Yes	--	No
9.	crucible-giga-cu-177	10.64.164.177	Data Collection Unit	CUG7			12.1.2 r465	10.4.31	el8	n/a	2941	41	Yes	--	No
10.	crucible-giga-cu-178	10.64.164.178	Data Collection Unit	CUG8			12.1.2 r465	10.4.31	el8	n/a	2941	42	Yes	--	No
11.	crucible-giga-cu-179	10.64.164.179	Data Collection Unit	CUG9			12.1.2 r465	10.4.31	el8	n/a	2941	43	Yes	--	No
12.	crucible-giga-cu-180	10.64.164.180	Data Collection Unit	CUG10			12.1.2 r465	10.4.31	el8	n/a	2941	44	Yes	--	No
13.	crucible-giga-cu-191	10.64.164.191	Data Collection Unit	CUG11			12.1.2 r465	10.4.31	el8	n/a	2941	45	Yes	--	No

Convert PowerFlow and Restorepoint to OL8

See *Converting PowerFlow to Oracle Linux 8 (OL8)* in the PowerFlow documentation.

Convert Restorepoint to OL8

See *Converting Restorepoint to Oracle Linux 8* in the Restorepoint documentation.

Set Up New Features

For an overview of the new features in Oracle Linux 8, see Oracle Linux 8 Features and Benefits.

System Maintenance

For information about recommended maintenance schedules, see *Monitoring and Maintaining SL1* in the SL1 Product Documentation.

For information about applying updates and patches, see *The System Updates Page* in the SL1 Product Documentation.

ScienceLogic Support

To work with ScienceLogic Support, go to the ScienceLogic Support Site and click **[Submit a Case]** (username and password required).

If you need to register and set up an account, go to the Getting Started page at the Support site.

Chapter

4

Frequently Asked Questions (FAQs)

Overview

Below is a list of common questions and their answers related to the Oracle Linux 8 (OL8) conversion process.

> **CAUTION:** All customers must upgrade to SL1 version 12.1.x or later and convert to OL8 by October 31, 2024, or before upgrading to SL1 version 12.2.0. **If you take no action, all SL1 systems with OL7 will continue to run, but ScienceLogic will not support them, and the systems may not be secure.**

This chapter covers the following topics:

General

Why should I convert to OL8?

Oracle Linux 7 (OL7) will be going end-of-life (EOL) in 2024. The conversion to OL8 enables SL1 platform users to take advantage of key next generation technology to improve application performance, security, scalability, and reliability. For more information, see [Oracle Linux 8 Features and Benefits](#).

Are there any features of SL1 version 12.1.2 that are not available to me unless the underlying OS is updated to OL8?

No. All SL1 12.1.2 application features are available regardless of the operating system.

How long will the conversion to OL8 take?

The entire process to convert to Oracle Linux 8 depends on several factors. The conversion process is based on how many databases you have, the amount of data, the network connection between databases (HA/DR), and how many collectors you have.

ScienceLogic recommends that you plan at least 1 day to convert the Database Servers, and at least 1-2 days to convert the collectors, depending on how many collectors need to be converted.

If I upgrade to SL1 12.1.2, will my Database Server (CDB) convert to OL8?

No. If you upgrade to 12.1.2, you will have the benefit of new and updated SL1 features, but the operating system will remain on OL7. For new customers, a new install using 12.1.2 release code will be installed with OL8.

For on-premises platforms, how much downtime will there be during the conversion process?

The OL8 conversion process can be performed in several ways, depending on the environment and whether the SL1 appliance is being redeployed or performing an in-place conversion. The total downtime can be estimated via the following guidelines:

- **Single Database Server (CDB) configuration**: For the backup, redeploy, and restore process, estimate about 4-12 hours, depending on how much data is being restored.

- **Multiple Database Server (CDB) configuration**. For failover with the data replication resync process, expect very little downtime to perform the failover, typically less than 5 minutes per node. The data resync process runs in the background.

- **Collector Conversion**. For an in-place conversion, the time to complete the conversion is similar to performing an application upgrade. After the collectors have been upgraded to the SL1 12.1 release line, issuing the OL8 conversion script can be done for a single collector or in batches. The collector conversion is estimated to take up to 1 hour or more. Bulk conversions are done in parallel via separate threads, so there will not be any additional time.

- **Extended Architecture (EA)**:
  - **Compute Node (CN)**. The redeployment process can take up to 1 hour, assuming that all the prerequisites have been met.
  - **Storage Node (SN)**: Two options are available for the SN:
    - **Redeploy**: Up to one hour to redeploy all three Storage Nodes in the cluster, then the total time to restore the data, which is dependent on volume size. The current estimation is up to 24 hours to restore all the data across the cluster.
    - **In-place conversion**: The conversion process is run as a round robin for each node with a data sync process. This method is for large data volumes and has little to no down time.

For more information about the OL8 conversion steps for each environment, see *Step 2: Converting to OL8*.

Features and Benefits

What security benefits come with Oracle Linux 8?

Enhanced Security: Oracle Linux 8 comes with the latest security features and updates to ensure your system remains secure from potential threats. It includes new technologies like Kernel Runtime Guard (KRG) and control-flow enforcement technology (CET) that can provide additional layers of security to your infrastructure.

What performance benefits come with Oracle Linux 8?

Improved Performance: Oracle Linux 8 has been optimized for modern hardware architectures and comes with several performance enhancements such as support for NVMe drives, improved networking performance, and faster boot times. These improvements can help your business achieve better performance and responsiveness for your critical applications.

What platform stability benefits come with Oracle Linux 8?

Increased Stability: Oracle Linux 8 provides long-term support and stability, ensuring that your systems remain reliable and up to date. This can help reduce downtime and minimize the risk of system failures, providing increased business continuity and uptime.

What type of cost benefits can be realized with Oracle Linux 8?

Lower Total Cost of Ownership: Oracle Linux 8 is an open-source operating system, which means that it is freely available and does not require licensing fees. Additionally, it comes with built-in tools like DTrace and Ksplice that can help reduce maintenance costs and improve system reliability.

Pre-conversion Checks

Are there cases where my existing hardware will not support the conversion to OL8? And if so, how can I determine that during the conversion planning process?

Bare metal deployments running on older hardware may run into issues due to the OL8 operating system kernel and boot file system requirements. ScienceLogic has tested OL8 running on the Dell 450 and 650 hardware. Customers considering a technology refresh should contact ScienceLogic about minimum hardware specs requirements.

Are there any scenarios where the sizing (CPU, Memory, Disk size) of my current platform will not be optimal or sufficient after the OL8 conversion If so, is there a method to audit this during the conversion planning??

The minimum and recommended system requirements running Oracle Linux 8 are not different from Oracle Linux 7. Generally, all system requirements are driven by SL1 business needs based on device count and type, system processing driving data pull and config push services, PowerPacks, and automations.

The above statement is true for all newly installed (ISO or AMI) OL8-based deployments. For existing SL1 stacks upgrading to 12.1.x or later, the OL8 migration process might require additional disk space to complete the conversion. Use the following guidelines to determine disk space requirements for the OL8 migration script.

Disk Space Considerations

In some cases this reservation may have been used on the OL7 Source system, and the OL8 Destination will not have enough space for the migration to proceed. If this happens, there will be an error in Phase 2 of the migration indicating that there is insufficient space to match disk sizes for the migration process. It is possible to check if the space is sufficient by doing the following:

1. On the Source system, run the following command to see how much space is allocated to the **/data.local/db** logical volume and take note of the number:

   ```
   sudo lvdisplay /dev/em7vg/db | grep "Current LE"
   ```

2. On the OL8 Destination system, run the following commands to see how much space is allocation to the **/data.local/db** logical volume and how much free space is in the volume group. Take note of these numbers.

   ```
   sudo lvdisplay /dev/em7vg/db | grep "Current LE"
   ```

   ```
   sudo vgs --noheadings -o vg_free_count
   ```

3. Add up the numbers from step 2, and compare them to the number in step 1:

   ◦ If the total from step 2 is **greater** than the number from step 1, then you have sufficient space.

   ◦ If the total from step 2 is the **same or less** than the number from step 1, you will need to allocate more space to the OL8 Destination system.

The Conversion Process

Will I be able to convert to OL8 on my own?

Yes. This conversion guide will provide all the information you need if you prefer to perform the conversion yourself. Specific conversion details will be published as part of the SL1 12.1.x release.

How big will the patch upgrade file be for 12.1.x, considering it has both OL7 and OL8?

Due to the need to support both OL7 and OL8 patch upgrades, the SL1 patch bundle file could be as large as 6 GB. This will directly impact download times. Once downloaded, the patching service will upgrade using the appropriate binaries, so the staging and deploy process will remain unchanged.

Can I upgrade my collectors when 12.1.2 is released?

Yes, the process is to upgrade the SL1 platform and all collectors to the 12.1.2 release. At this point, SL1 will be running the 12.1.2 version on the Oracle Linux 7 operating system. You may continue to run in this configuration. When ready to complete the operating system conversion, the process is to migrate the core Database Server (CDB) or the Data Engine to Oracle Linux 8 first, then complete the collector conversion process after.

Converting the collector operating system can be accomplished one of two ways:

1. **Run the collector in-place OS conversion script**. This script converts the existing collectors to the OL8 operating system.

2. **Redeploy a new 12.1.2 collector**. Re-ISO the collector appliance to install a new collector running on Oracle Linux 8. This will require registering and connecting the new collector to the SL1 database.

Both options will replace the OL7 collector with a collector running the OL8 operating system. For more information, see *Message Collectors and Data Collectors*.

Should I simply redeploy my collectors?

Collectors can be treated a throwaway appliances where simply replacing them is the easiest and fastest option to install a new OL8 collector. This will require registering and connecting the collector to the SL1 database.

How does the collector conversion script work?

The OL8 collector conversion script is best suited for large collector networks for which you want to maintain the existing infrastructure. The process is designed to replace the existing collector appliance with the OL8 operating system and reapply all existing configurations. When the OL8 conversion is complete, the collector will reconnect to the database and resume data collection. The conversion script will allow you to convert one, many, or all collectors at a time.

ScienceLogic recommends that you run a test against a single collector first to get comfortable with the process. When you are ready to migrate the remaining collectors, you can select them in batches to complete the conversion process.

> **TIP:** To avoid data collection gaps, do not convert all collectors in a collector group at the same time.

For more information, see the section on converting *Message Collectors and Data Collectors*.

For the SL1 12.1.x release, with respect to "mixed-mode" (OL7 and OL8 appliances on the same stack), are there any combinations not supported?

In other words, can I install a collector from the ISO and run OL8 while my CDB has not yet been converted? Can I convert the Database Server to OL8 and support collectors that have not yet been converted?

The mixed-mode concept means running OL7 and OL8 appliances on the same stack. Unless a customer is starting a brand-new 12.1 ISO or AMI installation, which is all OL8-based, all existing customers upgrading to the 12.1.x release line can run in the mixed mode phase for a period. The most common mixed-mode scenarios are either:

- Installing a new 12.1.0.2 OL8 collector and registering it to a recently upgraded 12.1.0.2 OL7 distributed stack

- Completing the core database conversion process to get the Database Server or Data Engine to OL8, but have not started the collector conversion process yet

Mixed-mode will not be supported in the 12.2 release and beyond. This means all existing customers must upgrade to 12.1 and complete the OL8 conversion before upgrading to future major releases.

What is the minimum version of SL1 I need to be running to convert to OL8?

To convert SL1 to running on Oracle Linux 8, you must upgrade the SL1 to the 12.1.x release.

> **NOTE:** OL7 to OL8 conversions are only supported on the 12.1.x release line. Future major releases, such as 12.2, 12.3, and later, will only support OL8.

Can I convert to OL8 if I am using a version of SL1 before version 11.2 (such as 7.x, 8.x, or 10.x)?

You can convert to OL8 from an older version of SL1, but you will need to follow the steps covered in the *Upgrade Paths*, below.

If you are running a version of SL1 older than 8.12, see *Upgrading from Versions of SL1 Before 11.x*.

Where can I find the "silo-migrate" migration script?

The OL8 conversion scripts are all embedded in the 12.1.x release. After you upgrade SL1 to 12.1.x, you can access the scripts directly via the command line interface (CLI). For more information, see the different configuration steps in the *Converting to OL8* chapter.

Who is responsible for converting collectors to OL8?

Converting collectors to OL8 is the customer's responsibility, unless you are engaging ScienceLogic for assistance.

Do the Extended Architecture components need to convert to Oracle Linux 8?

Yes, the SL1 Extended Architecture (EA) platform utilizes the Oracle Linux operating system and will require that all the components to be migrated to OL8. For more information, see the different configuration steps for EA in the *Converting to OL8* chapter.

How do I convert an on-premises Compute Node (CN) to OL8?

Converting the Extended Architecture (EA) Compute Nodes to OL8 is a straightforward redeployment process. The process is the same procedure to upgrading to a new release version. Pointing the EA platform service version to 12.1.x will install the appropriate charts and containers as well as import the OL8 operating system files. For more information, see the different configuration steps for EA in the *Converting to OL8* chapter.

How do I convert an on-premises Storage Node (SN) to OL8?

There are two options to convert your Storage Node cluster to OL8.

1. Back up and restore using an AWS S3 bucket (ideal for smaller environments).

2. A rolling storage node conversion process (ideal for larger environments requiring minimal downtime).

For more information, see the different configuration steps for EA in the *Converting to OL8* chapter.

I have a lot of Storage cluster data. What are the best options to limit downtime?

The EA Storage cluster rolling conversion method is based on the principle that Scylla will maintain the cluster operationally while each individual storage node is re-ISOed. This round-robin method is tailored to clusters with large data sets or minimal downtime is required. The steps include:

1. Remove a node from the cluster.
2. Re-ISO the node with OL8 ISO files.
3. Re-add the Scylla node.
4. Verify the node is back online attached to the cluster.
5. Repeat these steps for each Storage Node.

For more information, see the different configuration steps for EA in the *Converting to OL8* chapter.

How do I convert an on-premises Management Node (MN) to OL8?

To convert the Management Node to OL8, you will need to run a re-ISO process to install the OL8 operating system. This can be accomplished as part of the normal SL1 upgrade process.

I'm running the Extended Architecture in AWS. Do I have to convert the EKS Compute cluster to OL8?

The AWS Compute Node cluster is managed by AWS Elastic Kubernetes Service (EKS) and does not require an OL8 conversion. With most SL1 Extended Architecture upgrades does include updating the Kubernetes to the latest service version. When upgrading SL1 to the 12.1.x release version, there will be a step to update Kubernetes. For more information, see *Extended Architecture (AWS)*.

What is the process to convert my AWS Storage cluster to OL8?

The Extended Architecture (EA) Storage Node cluster running in AWS has two options to convert the cluster to Oracle Linux 8. They include:

1. Back up and restore using an S3 bucket (ideal for smaller deployments).
2. OL8 Blue/Green redeploy using a secondary data center

For more information, see *Extended Architecture (AWS)*.

In what circumstances is it better to upgrade collectors before migrating the Database Server to OL8?

Upgrade your collectors first if you need to refresh collectors anyway and you want to spread out the conversion process. If you only have a few collectors installed, or the collector network requires a tech refresh, you might want to simply install new collectors instead of converting the collectors.

What is the preferred order to convert SL1 and collectors?

The preferred order of operation is as follows:

1. SL1 application upgrade to 12.1.X using the **silo_upgrade** patch bundle.

2. Complete the core Database Server (CDB) or Data Engine (DE) OL8 conversion.

3. Initiate the collector conversion process.

Converting PowerFlow to OL8

When can I convert my PowerFlow system to OL8?

You can convert PowerFlow to OL8 starting in early Q2 2024.

What if my PowerFlow system is running OL7, but my SL1 system is running OL8 (or vice versa)?

SL1 and PowerFlow run on separate platforms, and there is no requirement that both SL1 and PowerFlow are on the same version of Oracle Linux. As a result, after you convert your SL1 system to OL8, all PowerFlow instances still operating on OL7 will continue to function normally. You can also have a PowerFlow system running OL8 and an SL1 system running a supported version of SL1 that is still on OL7. For the best results, however, ScienceLogic recommends that you convert both PowerFlow and SL1 to OL8 as soon as possible.

How will I convert my PowerFlow system to OL8?

To convert PowerFlow to OL8, you will need to run an upgrade script from ScienceLogic that is available with the ISO file for the PowerFlow release in early Q2 2024. Detailed upgrade information will be available in the release notes and manual for the PowerFlow Platform for that release.

PowerFlow Supported Operating Systems by Release

	Use Case	PowerFlow before 2.0	PowerFlow 2.1	PowerFlow 2.2	PowerFlow 2.x	PowerFlow 3.0+
1	Oracle Linux 7 Platform	Supported	Supported	Supported	Supported	Not supported
2	Oracle Linux 8 Platform	Not supported	Not supported	Not supported	Not supported	Supported

All PowerFlow customers upgrading to PowerFlow version 3.0 during the first half of 2024 will automatically be converted to running on the Oracle Linux 8 operating system.

Converting Restorepoint to OL8

What is the plan for converting Restorepoint to OL8? Will mixed mode support include Restorepoint on CentOS or OL8 with SL1 at OL8?

Depending on the Restorepoint version, the application currently runs on either CentOS or OL8. There are no dependencies between Restorepoint and SL1. For more information about converting Restorepoint to OL8, see *Converting Restorepoint to Oracle Linux 8* in the Restorepoint documentation.

The Restorepoint release is completely independent of the SL1 platform. For more information, see *RestorepointSupported Operating Systems by Release,* below.

> **NOTE:** If Restorepoint is installed on the SL1 collector, you will need to re-install Restorepoint after the SL1 collector is converted to OL8.

Restorepoint Supported Operating Systems by Release

	Use Case	Restorepoint before 5.0	Restorepoint 5.4	Restorepoint 5.7	Restorepoint 5.x	Restorepoint 6.0+
1	CentOS Platform	**Supported**: Existing customers running CentOS operating system (no near-term EOL milestone)	**Supported**: Existing customers upgrading remain on CentOS	Not supported	Not supported	Not supported
2	Oracle Linux 8 Platform	**Not Supported**: Older RP versions did not support OL8	**Supported**: Customers installing 5.4 are deployed running OL8	**Supported**: Customers installing 5.7 are running OL8	**Supported**: Customers installing 5.7 are running OL8	**Supported**: Customers installing 5.7 are running OL8

All Restorepoint customers upgrading to Restorepoint version 5.7 during the first half of 2024 will automatically be converted to running on the Oracle Linux 8 operating system.

Converting Skylar Automated RCA to OL8

What is the plan for converting Zebrium to OL8? Will mixed-mode support include Zebrium with SL1 at OL8?

Zebrium runs on a micro-service architecture is completely independent of the SL1 platform. The operating system that SL1 is running has no impact on the Zebrium application.

Converting Other Applications to OL8

What are the impacts regarding CrowdStrike running on the SL1 collector after the collector is converted to OL8?

If the CrowdStrike application is installed on the SL1 collector, you will need to re-install CrowdStrike after the collector is converted to OL8.

What are the impacts regarding CyberArk running on the SL1 collector after the collector is converted to OL8?

If the CyberArk application is installed on the SL1 collector, you will need to re-install CyberArk after the collector is converted to OL8.

What is the plan for MUD deployments (as we only have MUD on v11.3)?

The MUD STIG release with Oracle Linux 8 is under development. It is targeted for release in Q1 2024 as part of the 12.2.x release line.

Technical

Are there any kernel improvements that improve platforms running on Oracle Linux 8?

Oracle Linux 8 comes with a new, optimized kernel that provides better performance, scalability, and stability. The kernel includes several new features, such as Control Flow Integrity (CFI) and Live Kernel Patching, which can help improve system security and reduce downtime.

What Oracle Linux 8 improvements are there for running container-based services?

Oracle Linux 8 includes the latest container tools and technologies, such as Docker and Kubernetes, to provide a modern container environment. This can help improve application portability, scalability, and reliability.

Are there any Oracle Linux 8 improvements with file system support?

Oracle Linux 8 supports the latest file systems, including XFS and Btrfs, which provide improved performance, scalability, and data protection. The file system enhancements can also help improve the efficiency of your storage infrastructure.

Does Oracle Linux 8 have any administration management improvements?

Oracle Linux 8 comes with improved management tools, including the Cockpit web console and the Oracle Linux Web Console, which provide an intuitive, web-based interface for managing your Linux systems. These tools can help simplify system administration and improve productivity.

Does Oracle Linux 8 run on the latest hardware platforms?

Oracle Linux 8 is optimized for the latest hardware architectures, including Intel and AMD processors. It supports advanced hardware features like NVMe drives and RDMA networking, which can help improve system performance and scalability.

What data backup protection decisions do I need to consider before converting to Oracle Linux 8?

Backing up your data: Before starting the migration process, make sure to back up all important data, including databases, applications, and configuration files. This will ensure that you have a copy of your data in case anything goes wrong during the migration.

Should a data encryption service be used during the Oracle Linux 8 conversion?

If your data contains sensitive or confidential information, consider using encryption to protect it during the conversion. You can use tools like dm-crypt or LUKS to encrypt your data at rest, or use SSL or TLS encryption to protect data in transit.

What secure transfer protocols should be considered during the Oracle Linux 8 conversion?

When transferring data between systems, use secure transfer protocols like SCP or SFTP to protect your data from interception or tampering.

After completing the Oracle Linux 8 conversion, how should I verify data integrity?

After transferring your data to the new system, verify its integrity to ensure that it has not been corrupted during the transfer process. You can use tools like md5sum or sha256sum to calculate checksums for your data and compare them to the original checksums to ensure that the data has not been modified.

What Oracle Linux 8 post-conversion steps should I take?

After migrating your data to the new system, test your applications thoroughly to ensure that they are working correctly and that your data is accessible and usable. Make sure to test all critical functionality and perform load testing to ensure that your applications can handle the expected workload. For more information, see *Step 3: After You Convert to OL8*.

SaaS

For my SaaS platforms, how much down time will there be during the conversion process?

The conversion to Oracle Linux 8 for SaaS customers will include very minimal downtime. The ScienceLogic SRE team will coordinate with SaaS customers to schedule the OL8 migration maintenance period. SaaS customers will be required to plan their collector conversion with the Customer Service Managers (CSM) so that the SRE team can plan for the collector conversion period.

Who is responsible for upgrading my SaaS stack components?

The ScienceLogic SRE team is responsible for coordinating and converting the AWS SaaS components. The customer is responsible for converting the edge collectors. The SRE team will be in contact with each SaaS customer to coordinate the conversion date and time.

Security

Is SL1 running Oracle Linux 8 securely?

Yes. All instances of SL1 are secure. Moving to Oracle Linux 8 updates the SL1 operating system to the latest technology, ensuring platform security.

Is SL1 running Oracle Linux 8 Department of Defense information networks (DoDIN) certified?

Like previous versions, SL1 will be tested and certified to meet DoDIN APL standards. Because ScienceLogic is changing the core operating system, it will require full DoDIN testing. The 12.1.x Golden Gate release is

currently not DoDIN-certified. The security team will be submitting the paperwork to initiate the certification process later in 2023. An announcement will be made for existing DoDIN customers when upgrades are certified.

Is there an SL1 Military Unique Deployment (MUD) version?

The 12.1.x Golden Gate release version is not MUD-certified. Additional work is required to complete the development and testing of all the MUD scripts.

If I am a government customer using the (MUD), how will the conversion affect me?

The hardened MUD version will be aligned with the 12.2.1 release, which is targeted for the first quarter of 2024 (Q1 2024). If you need a MUD version, you can start converting at that point. However, if a federal customer is tied to DoDIN, there are additional DoDIN certification approvals before federal customers are authorized to upgrade to the 12.2 release line.

Does SL1 running Oracle Linux 8 support the Federal Information Processing Standard (FIPS)?

As part of the 12.2.1 release, MUD scripts with FIPS will be enabled, as with previous MUD versions.

What is the minimum Transport Layer Security (TLS) version supported?

The TLS version can be set to 1.0, 1.1, and 1.2, with TLS 1.2 as the default setting.

How will the conversion from Python 2.x to Python 3.x impact the OL8 conversion process?

With the introduction of Python 3 support in SL1 11.3.0, ScienceLogic announced its intent to deprecate support for Python 2 in a future release. At that time, any custom Python code that you have written within SL1 will cease to work properly. Therefore, if you currently use custom Python 2 code, including custom PowerPacks, ScienceLogic strongly recommends that you proactively convert it to use Python 3.

Additional information about this change will be in the release notes and related documentation for the SL1 version in which Python 2 support is deprecated. The ScienceLogic Support site also has additional information on the *Python 3 Resource Center* page.

SL1 Upgrade Paths

Enterprise Deployments (non-MUD)

The following table lists the upgrade paths to SL1 12.1.x for enterprise deployments that are non-Military Unique Deployment (non-MUD):

Upgrading from	Upgrade Path to 12.1.x or later				Target Release
7.x	8.12	10.2	11.3	12.1.0.2*	12.1.x
8.x	8.12	10.2	11.3	12.1.0.2*	12.1.x
10.1			11.3	12.1.0.2*	12.1.x
10.2			11.3	12.1.0.2*	12.1.x
11.1				12.1.0.2*	12.1.x
11.2				12.1.0.2*	12.1.x
11.3				12.1.0.2*	12.1.x
12.1				12.1.0.2*	12.1.x

* When upgrading to SL1 12.1.x, you will need a 30-minute pause on 12.1.0.2 so the upgrade procedure can import the RPM file for OL7 and OL8. No 12.1.0.2 staging or deployment is needed.

Federal and/or Government Deployments (MUD)

The following table lists the upgrade paths to SL1 12.1.x for federal and/or government deployments (MUD):

Upgrading from	Upgrade Path to 12.2.1 or later				Target Release
7.x	8.12	10.2	11.3	12.1.0.2*	12.2.1
8.x	8.12	10.2	11.3	12.1.0.2*	12.2.1
10.1			11.3	12.1.0.2*	12.2.1
10.2			11.3	12.1.0.2*	12.2.1
11.1				12.1.0.2*	12.2.1
11.2				12.1.0.2*	12.2.1
11.3				12.1.0.2*	12.2.1
12.1					12.2.1

* When upgrading to SL1 12.2.1, you will need a 30-minute pause on 12.1.0.2 so the upgrade procedure can import the RPM file for OL7 and OL8. No 12.1.0.2 staging or deployment is needed.

Interoperability Matrix

The interoperability matrix below represents the known supported SL1 operating system configurations and the respective mixed-mode (OL7 and OL8) use cases by release. This table lets you see what is feasible for each given release to help you plan your OL8 conversion strategy.

Distributed Architecture

	Use Case	SL1 12.1.0.2	SL1 12.1.x	SL1 12.2.0 SL1 12.2.1
1A	OL7 Core CDB/DE with OL7 collectors	**Supported**: Standard OL7 upgrade (silo_update bundle)	**Supported**: Standard OL7 silo_ update bundle. Requires consuming	**Not Supported for 12.2.0 or later**: You must complete OL8 conversion prior to 12.2 upgrade

	Use Case	SL1 12.1.0.2	SL1 12.1.x	SL1 12.2.0 SL1 12.2.1
			12.1.0.2 RPM file (imported by the upgrade procedure) before upgrading to 12.1.x	* Pre-upgrade check validation
1B	OL7 Core CDB/DE with OL7 collectors + adding new OL8 collector (12.1)	**Not Applicable**: No mixed-mode	**Not Recommended**: OL7 core with adding new OL8 collectors. This mixed-mode combination is technically feasible, but not recommended. ScienceLogic will provide support, but this is not a standard nor a recommended configuration.	
1C	OL7 Core CDB/DE with OL7 collectors + adding new OL7 collector (11.3)	**Not Applicable**: No mixed-mode	**Not Recommended**: OL7 core with adding new OL7 collector. After upgrading to 12.1.X adding a new OL7 collector will require installing an 11.3 collector first, then upgrading. This is technically feasible but not recommended.	
2	OL8 Core CDB/DE with OL8 collectors	**Supported**: New install configuration	**Supported**: Direct install AMI (SRE). Existing on-premises customers install 12.1.0.2 ISO first, then patch update to 12.1.x	**Supported**: All customers must install new or complete OL8 conversion first before upgrading to 12.2 release
3	OL8 Core CDB/DE with OL7 collectors	**Not Applicable**: No mixed-mode	**Supported**: Primary use case to convert core CDB/DE to OL8 first. At that point, OL7 collectors are connected to OL8	**Not Supported for 12.2.0 or later**: You must complete OL8 conversion prior to 12.2 upgrade * Pre-upgrade check validation

	Use Case	SL1 12.1.0.2	SL1 12.1.x	SL1 12.2.0	SL1 12.2.1
			database		
4	OL8 Core CDB/DE with OL7 and OL8 collectors	**Not Applicable**: No mixed-mode	**Supported**: Primary use case after converting core CDB/DE to OL8 and starting collector migration; will cause several OL7 and OL8 collectors to be connected to an OL8 database		

Distributed Architecture: MUD

	Use Case	SL1 12.1.0.2	SL1 12.1.x	SL1 12.2.0	SL1 12.2.1
5	OL7 Core CDB/DE with OL7 collectors	**MUD Not Supported**: OL8 STIG/SRG not planned		**MUD Not Supported**: Work underway for 12.2.X release	**Supported**: Planned MUD work completed for DoDIN certification
6	OL7 Core CDB/DE with OL7 collectors + adding new OL8 collector				
7	OL7 Core CDB/DE with OL7 collectors + adding new OL7 collector (11.3)				
8	OL8 Core CDB/DE with OL7 collectors				
9	OL8 Core CDB/DE with OL7 and OL8 collectors				

Extended Architecture (EA)

	Use Case	SL1 12.1.0.2	SL1 12.1.x	SL1 12.2.0	SL1 12.2.1
10	OL7 Distributed with OL7 Extended architecture	**Supported**: Primary supported configuration	**Not Supported**: OL7 distributed architecture is allowed after upgrade, but EA will redeploy as OL8 during upgrade	**Not Supported for 12.2.0 or later**: You must complete OL8 conversion prior to 12.2 upgrade * Pre-upgrade check validation	
11	OL8 Distributed with OL7 Extended architecture	**Not Recommended**: OL8 distributed running OL7 EA:	**Not Supported**: OL8 distributed architecture is		

	Use Case	SL1 12.1.0.2	SL1 12.1.x	SL1 12.2.0	SL1 12.2.1
		Technically feasible, but not recommended	allowed after conversion, but EA will redeploy as OL8 during upgrade		
12	OL8 Distributed with OL8 Extended architecture	**Supported**: For new installations only	**Supported**: OL8 Distributed conversion completed and EA redeployed as OL8	**Supported**: OL8 Distributed conversion completed and EA redeployed as OL8	

For more information about configuring many of these use cases, see *Converting to OL8*.

For Additional Questions

I have additional questions. How do I get further help?

Please contact your ScienceLogic Customer Success Manager (CSM) for additional information.

ScienceLogic