Skylar One Oracle Linux 9 Conversion Guide

Skylar One version 13.0.6

Table of Contents

Chapter 1

Oracle Linux 9 Conversion Guide

Overview

As part of the Skylar One platform life cycle, ScienceLogic is upgrading the platform operating system from Oracle Linux 8 (OL8) to Oracle Linux 9 (OL9). This guide covers all conversion paths and procedures for moving Skylar One appliances from OL8 to OL9.

Conversion to OL9 is available with Skylar One version 13.0.6 and later.

Skylar One ships as a controlled platform runtime with its own tested dependency and update life cycle. OL8 to OL9 conversion runs through image-based tooling workflows that ScienceLogic can fully validate and reproduce. These workflows include the **silo-migrate** process for the core Database Server (CDB) and the **recovery-image** plus coordinator service for Skylar One collectors.

The OL9 conversion is split into two tracks, based on the Skylar One appliance type:

- Core CDB and Database Server appliances, including All-In-One Appliances, standalone Database Servers, and High Availability (HA), Disaster Recovery (DR), and High Availability with Disaster Recovery (HA+DR) appliances. These appliances convert using the **silo-migrate** tooling, extended for OL8 to OL9.
- Skylar One collectors convert through the **recovery-image** flow, orchestrated by the **distribution-upgrade** coordinator service and triggered from the Skylar One user interface.

> CAUTION: To avoid outages or down time, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL9 *before* attempting to convert a production system.

Workflow for Converting Skylar One to Oracle Linux 9

A typical OL8 to OL9 conversion for Skylar One includes the following steps:

1. *Review the Key Architectural Points for OL9*.
2. For Central Database Server (CDB) and Database Server appliances: *complete pre-conversion checks, confirm staging, and run the silo-migrate OL8-to-OL9 procedure for your topology*.

> **NOTE:** For Skylar One collectors, the only process currently to "upgrade" collectors to OL9 right now is to re-ISO them. A future version of Skylar One will have an in-place re-imaging option

3. For Federal STIG deployments in their own AWS account: *upgrade the Data Engine* as needed.
4. Complete post-conversion steps using the *Validation section of the Conversion Phases for the Core CDB* and the Validation and refresh section of the Conversion Phases for the Collectors. Retire or decommission OL8 source nodes after validation.

> **NOTE:** OL9 uses a distinct image policy from OL8. Eligibility and conversion progress for each appliance can be tracked independently in the Skylar One user interface and in the Skylar One internal tables.

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

> **TIP:** For more information, see the *Frequently Asked Questions (FAQs).*

Chapter 2

Reviewing Key Architectural Points

Overview

ScienceLogic strongly recommends that you review the architectural facts for the Oracle Linux 8 (OL8) to Oracle Linux 9 (OL9) conversion model in this chapter before you move on to the procedure sections.

> **CAUTION:** To avoid outages or downtime, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL9 before attempting to convert a production system.

This chapter covers the following topics:

Mixed OL8 and OL9 Clusters Are Supported

Corosync and Pacemaker versions on OL8 and OL9 are compatible. This is the most significant difference from the OL7-to-OL8 conversion model.

OL9 nodes can join an existing OL8 cluster, which enables a rolling, in-cluster replacement pattern for HA, DR, and HA+DR topologies. The fully detached migration pattern is no longer the only option.

Image-Based, ScienceLogic-Controlled Delivery

The OL9 appliance state ships through ScienceLogic-controlled images, managed through the platform's existing image policy and staging tables, which includes:

- master_platform.image_policy
- image_staging_policy
- appliance_image_status
- appliance_image_staging

Conversion is a platform life-cycle event, not a host-managed OS upgrade.

Clear Support Boundary

The Skylar One runtime remains an isolated, controlled platform component before, during, and after conversion.

Customer-managed host responsibilities, including domain join, CAC/PIV, host scanning, EDR, and host-based access controls, live on the surrounding host environment, not inside the runtime. The OL9 conversion preserves that separation.

NOTE: Conversion tooling is validated against the ScienceLogic-supplied OL8 source state and the ScienceLogic-supplied OL9 target state. Conversions of modified runtimes are out of the scope of this document.

Chapter 3

Converting the Core CDB and Database Server Appliance

Overview

`silo-migrate` was the tooling used for OL7 to OL8 appliance migrations and is no longer the primary path for OS version changes on Skylar One appliances. For clustered topologies, the preferred tool is `silo-cluster-replace-node`, a guided node replacement workflow that is OS-version agnostic and applies equally to hardware failure recovery, planned refresh, VM migration, and OS upgrades.

For non-clustered topologies, the supported path is to back up and restore to a fresh OL9 appliance.

> **CAUTION:** To avoid outages or downtime, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL9.

This chapter covers the following topics:

Topologies and Conversion Methods

The conversion method depends on whether the appliance participates in an Skylar One cluster.

Clustered Topologies (HA, DR, HA+DR)

HA, DR, and HA+DR appliances use `silo-cluster-replace-node`. The OL9 replacement node is provisioned with the same hostname and IP addresses as the node being replaced and joins the existing OL8 cluster.

Corosync and Pacemaker are compatible across OL8 and OL9, so mixed-OS clusters are fully supported. Customers can run mixed for as long as needed and fail over to OL9 on their own schedule.

Non-clustered Topologies (All-In-One Appliance, Standalone Database Server)

All-In-One Appliance and standalone Database Server Appliances do not have cluster infrastructure (no Corosync, Pacemaker, or DRBD) and cannot use the node replacement workflow.

The supported conversion path for these topologies is to backup and restore to a fresh OL9 appliance.

Standalone Database Server: Convert to a Clustered Stack (Optional)

A standalone Database Server appliance can be configured for clustering before the OS conversion, following the standard process documented in the ScienceLogic public documentation.

After clustering is established, the appliance becomes an HA, DR, or HA+DR topology and an OL9 secondary can join the cluster directly, without a full backup and restore cycle.

When choosing to make a cluster from an existing standalone Database Server, the new nodes can be OL9. After the cluster is up, the standard node replacement workflow applies to the old standalone node. This is worth considering for any standalone Database Server where redundancy is planned in any case, as it eliminates the backup and restore step from the OS upgrade path.

Topology	Conversion Method	Mixed-OS in Cluster
All-In-One Appliance (AIO)	Back up and restore to OL9 appliance	N/A
Standalone Database Server	Back up and restore to OL9 appliance	N/A
Standalone Database Server (convert to cluster)	`silo-cluster-replace-node`	Yes
HA	`silo-cluster-replace-node`	Yes

Topology	Conversion Method	Mixed-OS in Cluster
DR	`silo-cluster-replace-node`	Yes
HA+DR	`silo-cluster-replace-node`	Yes

Restore Issue for Skylar One Systems with EKMS Enabled

The Enterprise Key Management Service (EKMS) feature is enabled by default for Skylar One versions 12.2.0 through 12.2.6. EKMS is disabled by default on 12.1.x and 12.3.1 and later. The default configuration only applies to new installations; if you are on a version where EKMS was enabled and you then update to a version where it is disabled by default, EKMS will still be enabled.

If you are restoring a full backup of an OL8 Skylar One system with EKMS enabled, you will need to run the following procedure as part of the OL9 conversion.

> **NOTE:** If this issue is encountered in AWS using RDS as a database, this issue could impact the user interface of both the active and passive Data Engines (DEs). The workaround for this issue will vary slightly for your configuration. Most on-premises customers will not be using DEs with RDS, so this issue would only impact the active node where the Database Server is running and serving the user interface. The workaround below has been generalized so it can be implemented on either configuration.

The following procedure will force EKMS to reinitialize to regenerate the files. Run the command on the affected node where **sl_vault** is not running and do ***not*** drop the user being used by the working node. This is specific to RDS being used; if RDS is not being used, there is no way for there to be a working node if this issue is occurring.

You can find the working user in the **/tmp/vault_conf.yml** directory of the working node. Do not delete the working user from the database when applying this workaround.

If both the Active and Passive DEs are not working, run the workaround on both DEs, but note that running the command to drop the *em7-security* users on both nodes will be redundant, as the users will be dropped from RDS when it is run on the first node.

To address this issue for an RDS installation:

1. If you are on a FedRAMP deployment, or similarly hardened deployment where the db permissions are locked down, make sure the `dbuser` under the `[LOCAL]` section in **silo.conf** has the right permissions. Otherwise, you may skip the following bullets.

    - Use `visilo` to view the **silo.conf** file to find the `dbuser`.

      ```
      silo_mysql -e "SELECT user, grant_priv FROM mysql.user WHERE
      user = '<dbuser>'"
      ```

      If the response is not `Y`, ask your database administrator to grant you the permissions for that user.

2. Stop the EKMS services and mask the EKMS services to keep them from restarting when EKMS reinitializes:

```
sudo systemctl stop sl_vault sl-vaultmngt
```

```
sudo systemctl mask sl_vault sl-vaultmngt
```

3. Delete the users. It should regenerate when EKMS reinitializes if necessary. Check the working node to see which user it is using in grep username **/tmp/vault_conf.yml** and drop the one that is not in use by running the following commands:

```
silo_mysql -e "DROP USER IF EXISTS 'em7-security'"
```

```
silo_mysql -e "DROP USER IF EXISTS 'em7-security-02'"
```

To address this issue for a non-RDS installation:

1. Run the grep on the Core Database Server (CDB) and drop the user that is not returned.
2. Remove the bad EKMS files:

```
sudo rm -f /tmp/vault_conf.yml /etc/sl_vault/vault_conf.yml /etc/sl_
vault/encryption_key /opt/em7/services/sl_vault/config/hcl/vault.hcl
```

3. Copy the default vault configuration template for generating:

```
sudo cp -v /opt/em7/services/sl_vault/utils/vault_conf.yml /etc/sl_
vault/
```

4. Set permissions for the vault configuration:

```
sudo chown s-em7-security:s-em7-security /etc/sl_vault/vault_conf.yml
```

5. Re-enable the EKMS services:

```
sudo systemctl unmask sl_vault sl-vaultmngt
```

```
sudo systemctl start sl_vault sl-vaultmngt
```

Node Replacement Workflow

For clustered topologies, the OL8 to OL9 conversion runs through the standard node replacement workflow. The same `silo-cluster-replace-node` command is run on two systems in sequence, and the script self-detects which step to perform based on the state of the node it executes on.

Configuration sync is handled inline during the replacement install, and DRBD replicates data automatically once the node joins.

For more information, see Updating the Nodes in a CDB Cluster.

Pre-check

Before starting, run `silo-cluster-replace-node --check` on the currently active cluster node. This is a non-destructive pre-flight validation that can be run at any time.

On the active node the pre-check validates:

- That this is the current active node
- That the local cluster configuration matches the database
- Node identity (hostname and IP addresses match the expected slot)
- That the Pacemaker CIB template is current
- TOTAL_LE drift, and offers to correct it
- DRBD peer connection and disk state

The same `--check` flag can be run on the replacement node before it joins. In that context it validates OS-level prerequisites and, if the active node's IP is provided, retrieves the cluster configuration and checks that the replacement node's hostname and IP addresses match the expected slot.

The appliance should also pass standard Skylar One self-monitoring health checks and show a clean overall status before beginning.

Step 1. Eviction (run on the currently active cluster node)

Run `silo-cluster-replace-node` on the currently active cluster node while the OL8 node being replaced is offline. The wizard:

- Validates active cluster status
- Displays the current cluster layout
- Requires the operator to confirm the offline node has been shut down and that the replacement will use the same hostname and IP addresses
- Presents offline DRBD peers for selection
- Removes the selected node from Pacemaker cluster membership

Before making any changes, the wizard automatically triggers `silo-config-sync save` to snapshot current cluster configuration to the database.

Step 2. Replacement node install (run on the new OL9 node)

Run `silo-cluster-replace-node` on the OL9 replacement node. The script:

- Retrieves the full cluster configuration from the active node's database
- Generates Corosync, DRBD, and Pacemaker configuration files
- Resizes the local storage volume to match the cluster's established DRBD allocation
- Installs cluster services and the node joins the cluster

During the install, the script interactively prompts the operator to optionally restore pre-replacement configuration snapshots (if available for this node's licensed IP address) and to optionally sync current configuration from the active node via `silo-config-sync`. Both steps are optional.

DRBD data sync

DRBD replicates all data from the active node automatically after the replacement node joins. Monitor progress with `drbdadm status`. Do not proceed to cutover until all peers show `UpToDate`.

Cutover

Once DRBD sync completes, promote the OL9 node to active following standard cluster failover procedures.

OL8 node retirement

Once the OL9 node is confirmed active and healthy, the node replacement procedure can be done for any remaining OL8 nodes.

Configuration and Identity Preservation

The replacement workflow preserves system configuration and appliance identity through `silo-config-sync`. During *Step 1 (eviction)*, all monitored configuration files are snapshotted to the cluster database via `silo-config-sync save`. During Step 2 (replacement node install), the operator is prompted to sync those files to the new node.

Per-file merge strategies ensure that node-specific values (this node's IP address, cluster HA configuration, and local encryption keys) are never overwritten from the source. Credential fields are re-encrypted with the replacement node's local key automatically. PhoneHome identity and registration state carry forward through this process, so downstream connectivity does not need to be re-established after conversion.

When the customer environment changes IP, NAT, or PhoneHome destination during the conversion window, the standard phonehome update steps apply.

Post-Conversion Validation

The appliance should pass standard Skylar One self-monitoring health checks and show a clean overall status.

Full post-conversion checklist:

- License state confirmed on the OL9 active node.
- MariaDB running, schema and replication state healthy.
- `system_status.sh` returns clean across the appliance.

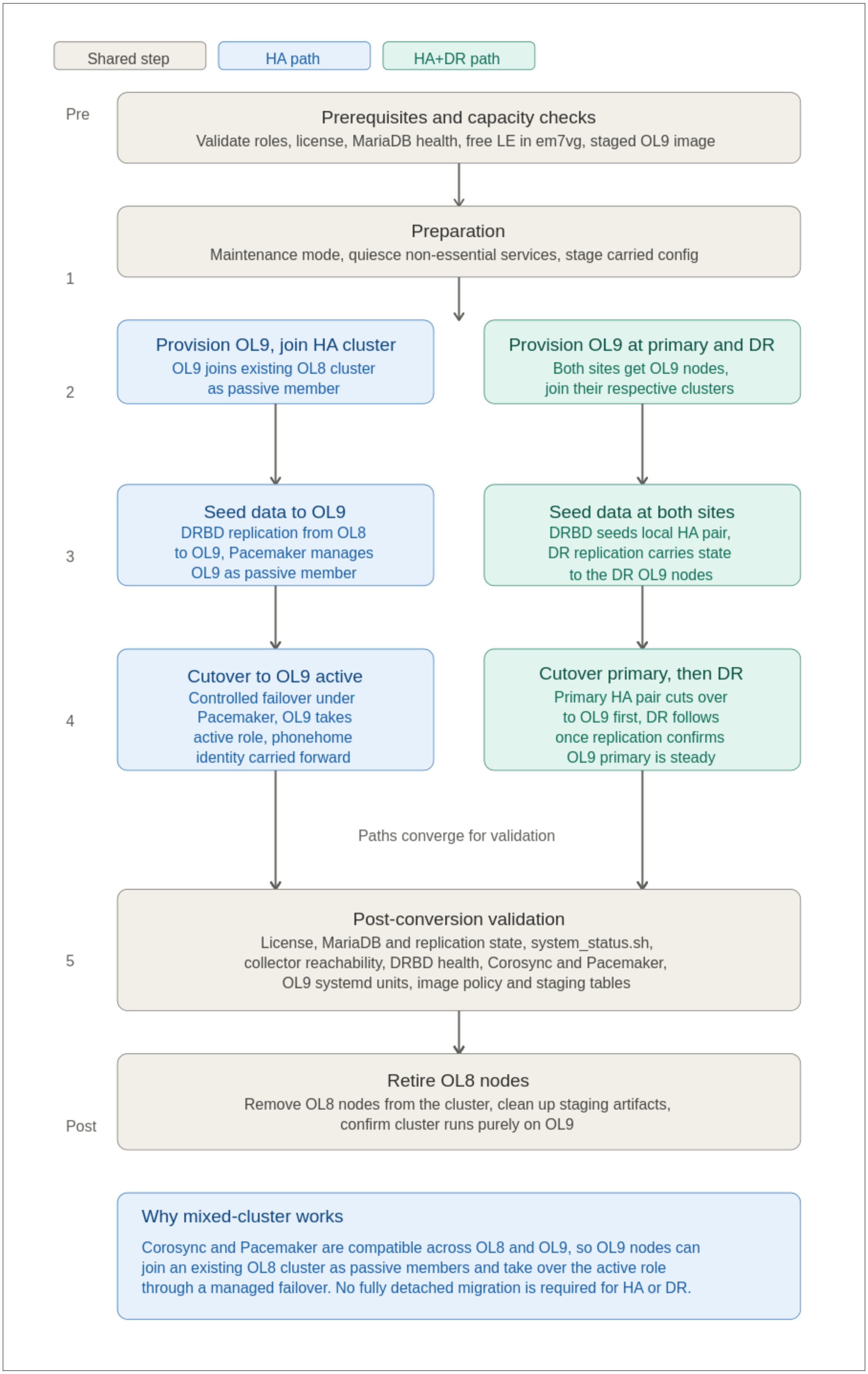
- Collector reachability and data flow verified end-to-end.
- DRBD status healthy – all peers `UpToDate`.
- Corosync and Pacemaker status healthy. Failover and fence behaviors validated.
- OL9-specific systemd units present and active.
- OL9 security and STIG baseline expectations validated (particularly relevant for federal accounts where the OL9 STIG baseline differs from OL8).
- Image policy and staging tables (`master_platform.image_policy`, `appliance_image_status`, `appliance_image_staging`) reflect the OL9 state of the appliance.

Validation gate. For HA and HA+DR topologies, validation runs while OL8 nodes are still in the cluster as passive members. If anything fails, the operator can roll back to OL8 active without data loss. Once the OL8 node is retired, that rollback path is gone. Validation is the last clean exit point.

HA and HA+DR Conversion Workflow

The HA and HA+DR conversion follows the same workflow as the *Node Replacement workflow*, above, but the replacement provisioning, DRBD sync, and cutover steps differ between the two topologies.

Refer to following workflow for a diagram mapping each step across both HA and HA+DR topologies.



Shared step
HA path
HA+DR path
Pre
Prerequisites and capacity checks
Validate roles, license, MariaDB health, free LE in em7vg, staged OL9 image
1
Preparation
Maintenance mode, quiesce non-essential services, stage carried config
2
Provision OL9, join HA cluster
OL9 joins existing OL8 cluster as passive member
Provision OL9 at primary and DR
Both sites get OL9 nodes, join their respective clusters
3
Seed data to OL9
DRBD replication from OL8 to OL9, Pacemaker manages OL9 as passive member
Seed data at both sites
DRBD seeds local HA pair, DR replication carries state to the DR OL9 nodes
4
Cutover to OL9 active
Controlled failover under Pacemaker, OL9 takes active role, phonehome identity carried forward
Cutover primary, then DR
Primary HA pair cuts over to OL9 first, DR follows once replication confirms OL9 primary is steady
Paths converge for validation
5
Post-conversion validation
License, MariaDB and replication state, system_status.sh, collector reachability, DRBD health, Corosync and Pacemaker, OL9 systemd units, image policy and staging tables
Post
Retire OL8 nodes
Remove OL8 nodes from the cluster, clean up staging artifacts, confirm cluster runs purely on OL9
Why mixed-cluster works
Corosync and Pacemaker are compatible across OL8 and OL9, so OL9 nodes can join an existing OL8 cluster as passive members and take over the active role through a managed failover. No fully detached migration is required for HA or DR.

What changes between HA and HA+DR

The pre-check, eviction, post-validation, and retirement steps run the same way regardless of topology. The split shows up in three places.

- **Replacement provisioning.** HA replaces one set of nodes in one cluster. HA+DR replaces nodes at both the primary and DR sites, each in their own cluster.
- **DRBD data sync.** HA: data replicates within the local HA pair. HA+DR: after the local HA pair syncs, the DR replication layer carries state to the DR-site nodes.
- **Cutover.** HA does one controlled failover using `silo-cluster-mgmt`. HA+DR cuts over the primary HA pair first, then brings the DR side over once the primary is stable.

Why post-conversion validation matters

Validation runs after cutover while OL8 nodes are still in the cluster as passive members. It is the gate that confirms the OL9 cluster is genuinely ready before the OL8 side is retired. The checklist in the *Post-Conversion Validation section* breaks into logical groups covering:

- **Platform identity and licensing.** License state has to be confirmed on the OL9 active node. The conversion carries the license forward, but validation makes sure it actually registered cleanly against the new node's identity.
- **Database health.** MariaDB needs to be running on the OL9 active node with a healthy schema. Replication state has to be clean, both for the local HA pair and (for HA+DR) for the DR replication link to the DR site.
- **Application-layer health.** `system_status.sh` has to return clean across the appliance. Collector reachability and data flow need to be verified end-to-end.
- **Cluster health.** DRBD status healthy where DRBD is still in use. Corosync and Pacemaker status healthy, and failover and fence behaviors validated.
- **OL9-specific items.** OL9 systemd units need to be present and active. Any OL9 security and STIG baseline expectations have to be validated, particularly relevant for federal accounts.
- **Platform state consistency.** Image policy and staging tables (`master_platform.image_policy`, `appliance_image_status`, `appliance_image_staging`) need to reflect the OL9 state of the appliance.

The silo-cluster-replace-node Script

The mechanics of taking an OL8 node out of a cluster and bringing the OL9 replacement in are handled by `silo-cluster-replace-node`. The script runs in two steps, executed on two different systems. The operator runs the same command both times, and the script detects which step to perform based on the state of the node it is executing on.

Step 1. Eviction (run on the currently active cluster node)

This step takes the offline OL8 node out of the cluster and gets the cluster ready to receive its OL9 replacement:

- Validates the node it is running on is the currently active node. Confirms the database is running, DRBD holds the Primary role, and cluster resources are active on this node.
- Verifies the local cluster configuration matches what is recorded in the cluster database.
- Checks the Pacemaker configuration is current against the Skylar One shipped template, and warns and offers to exit if an update is needed before proceeding.
- Displays the current cluster layout (architecture, node hostnames, IP addresses) for operator review.
- Requires the operator to explicitly confirm the node being replaced has been shut down per ScienceLogic documented procedure.
- Requires the operator to confirm the replacement node will use the same hostname and IP addresses as the node being removed.
- Detects and presents offline nodes through DRBD connection state for the operator to select.
- Requires a final confirmation before making any changes.
- Removes the offline node from Pacemaker cluster membership.
- Triggers a fresh snapshot of all cluster configuration files to the database via `silo-config-sync save`, so the replacement node can retrieve a current copy.
- Prints instructions for the operator to proceed on the replacement node.

Step 2. Replacement Node Setup (run on the new replacement node)

This step brings the OL9 replacement node into the cluster and starts DRBD replication:

- Confirms eviction was completed on the active node.
- Confirms the replacement node is configured with the same hostname and IP addresses as the evicted node.
- Retrieves the full cluster configuration from the active node's database (architecture, node IPs, DRBD settings, Pacemaker template).
- Generates Corosync, DRBD, and Pacemaker configuration files for this node.
- Resizes the local storage volume to match the cluster's established DRBD allocation.
- Installs and starts cluster services (Corosync, DRBD, Pacemaker).
- The node joins the cluster, and DRBD begins replicating all data from the active node automatically.
- Interactively prompts the operator to optionally restore pre-replacement configuration snapshots and to optionally sync current configuration from the active node via `silo-config-sync`. Both steps are optional; the replacement node is fully functional after the install regardless of whether either configuration step is performed.
- Validates the node has joined in the expected role.

- Provides instructions for monitoring replication progress and the cutover steps once sync is complete.

When both HA nodes are offline and the TERTIARY (DR) node is the only active node in the cluster, the script switches to a guided two-step rebuild workflow rather than running a standard single-node eviction.

The first replacement HA node installs in a modified mode. It initializes the HA ring and loads the cluster configuration, but holds resources offline until DRBD data sync from the TERTIARY completes and the operator manually promotes. Once the first HA node's sync is complete, the second HA node follows the standard single-missing-node replacement path.

Special case: HA+DR with both HA nodes offline (TERTIARY is sole active node)

When both HA nodes are offline and the TERTIARY (DR) node is the only active node in the cluster, the script switches to a guided two-step rebuild workflow rather than running a standard single-node eviction.

The first replacement HA node installs in a modified mode. It initializes the HA ring and loads the cluster configuration, but holds resources offline until DRBD data sync from the TERTIARY completes and the operator manually promotes. Once the first HA node's sync is complete, the second HA node follows the standard single-missing-node replacement path described above.

The TERTIARY detects this scenario automatically, so the operator does not need to invoke a different command. The script presents the modified workflow on its own.

Node Replacement Scope and Limitations

The `silo-cluster-replace-node` script covers the same-identity replacement case, which is the operating model behind the OL8 to OL9 conversion workflow described in *Node Replacement Workflow* and *HA and HA+DR Conversion Workflow*. The different-identity case is intentionally out of scope for self-service and is handled as a Support-assisted operation.

Non-clustered topologies (not supported by this workflow)

AIO and standalone DB appliances do not have the cluster infrastructure `silo-cluster-replace-node` requires. The script cannot be used on a non-clustered appliance. For non-clustered topologies, the supported conversion path is backup and restore to a fresh OL9 appliance. Standalone DB appliances that are configured for clustering before the conversion can use the node replacement workflow. For more information, see *Topologies and Conversion Methods*.

Same-identity replacement (supported, self-service)

The replacement node is provisioned with the exact hostname and IP addresses of the evicted node. Nothing about the running cluster's network configuration changes. The new node steps into the evicted node's slot.

This covers the large majority of real-world scenarios: hardware failure recovery, planned OS-version upgrades (including OL8 to OL9), and VM migration where the operator reprovisions with the same identity.

Different-identity replacement (deferred, Support-assisted)

Replacing a node with a new hostname, new IP addresses, or both is substantially more complex because it requires changes to the running cluster before the replacement node can join:

- The Corosync member list on all active nodes has to be updated with the new hostname.
- DRBD replication peer addresses on all active nodes have to be updated.
- Both services have to be reloaded non-disruptively while the cluster stays online.
- The cluster configuration database has to be updated with the new node identity, so subsequent operations use the correct values.
- All of this has to be sequenced correctly. Out-of-order execution can destabilize a running cluster.

The script detects this case and directs the operator to contact ScienceLogic Support rather than proceeding. Building the full self-service tooling for different-identity replacement is planned but deferred.

Why this limitation matters for OL8 to OL9 conversion

The *Replacement node install (run on the new OL9 node)* workflow and the *HA and HA+DR* workflow both assume same-identity replacement. If a customer's OL8 to OL9 conversion requires changing hostnames or IPs as part of the move (for example, consolidating into a different network segment), that work falls outside the self-service flow and needs to be sequenced with Support involvement.

Chapter

4

Upgrading Data Engines

Overview

This chapter covers upgrading Data Engine (Data Engine) nodes from OL8 to OL9 using a blue-green approach. The process spins up a new OL9 instance, migrates configuration from the current Data Engine, and performs a live cutover with minimal downtime.

Please note that these procedures are not meant for FedRAMP customers, but for customers who have Federal STIG deployments in their own AWS account.

> **CAUTION:** To avoid outages or downtime, ScienceLogic strongly recommends that you read through this guide and perform a test conversion to OL9 *before* attempting to convert a production system.

This chapter covers the following topics:

Prerequisites

- Ansible controller with access to AWS and both Data Engine nodes
- SOPS-encrypted secrets file at `{{ input_directory }}/secrets.enc.yml` containing `em7admin_password`
- SSH key access to Data Engine nodes as `em7admin`
- `ansible_become_pass` set in inventory for localhost

Gather Required Information

Before starting, collect the following information; the Skylar One version and build number must match between the AMI and siloupdate:

- **AMI name**: `em7_x86_64_12.5.4-25.el8-FullSTIG-DB-Remote`
- **Siloupdate**: `EM7_v12.5.4-25.Platform_2025-01-05.siloupdate`

Note that the version and build numbers (12.5.4-25) must be identical.

Upgrade Steps

1. To update all appliances to the target version, apply **siloupdate** to all classic appliances and verify with:

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

2. To set the cluster into maintenance mode, SSH to one of the Data Engines and run:

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

3. To identify the passive Data Engine, SSH to each Data Engine and check the HA status:

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

   Note which node is passive. The upgrade runs on the passive node first.

4. To run the deupgrade playbook, set the following environment variables and run the playbook. In the following example, `de02` is the passive node:

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

Example:
docker-compose -f docker-compose.external.yml run --rm \
-e REGION='us-east-1' \
-e CLUSTER='devstig' \
-e COMPONENT='de02' \
-e AMI='em7_x86_64_13.0.1-3.el9-FullSTIG-DB-Remote' \
deploy deupgrade
```

> **NOTE:** The Upgrade Steps *continue* after the following sub-topics.

Environment Variables

- `REGION`: AWS region of your cluster
- `CLUSTER`: Cluster name, which is used to locate the current Data Engine instance by tag
- `COMPONENT`: Component name of the passive Data Engine (**de01** or `de02`)
- `AMI`: AMI name corresponding to the target OL9 version and build

What the Playbook Does

1. Locates the current Data Engine by tag and collects its configuration.
2. Reads **silo.conf** (LOCAL, CENTRAL, HA, OS_HARDENING sections).
3. Copies all readable config files from the current Data Engine to the controller
4. Launches a new OL9 EC2 instance with user-data that sets the **em7admin** password and configures **silo.conf** with the database credentials for the current Data Engine.
5. Waits for SSH to become available on the new instance.
6. Copies all config files to the new Data Engine.
7. Transfers wildcard directories (`/etc/nginx/certs/`, MIBs)
8. Stops `phd` on the current Data Engine and starts it on the new Data Engine.

9. Moves the EIP (if present) from the current Data Engine to the new Data Engine.
10. Registers the new Data Engine with any existing ELB target groups

Upgrade Steps, continued:

5. When the playbook completes, verify the new OL9 node by going to the **Collector Status** page (System > Monitor > Collector Status) and checking that the collectors are available. You should also verify that all necessary services are running, and confirm portal access from the new node.
6. Promote the new node to active by connecting to the new OL9 node and running:

   ```
   silo_mysql -e "update sysinfo.system_heartbeat set appliance_id =
   <Data Engine Appliance ID>"
   ```

   You can find the Data Engine Appliance ID on the Skylar One **Appliances** page (System > Settings > Appliances). The new OL9 node is now active and the old OL8 node is passive.
7. Clean up the old node by deleting the old OL8 node from the **Appliances** page, and then you can terminate the old instance in AWS.
8. To upgrade the second Data Engine, repeat steps 3-7, above.
9. To remove maintenance mode, after both Data Engines are upgraded, SSH to one of the Data Engines and run:

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

Notes:

- The playbook handles environments with or without public IPs. EIP reassignment is skipped if no public IP is present on the current Data Engine.
- Config files that are unreadable or missing on the current Data Engine are skipped automatically.
- The `phd` service cutover is the point of live traffic handoff, so ensure the verification steps are complete before this point.

Chapter 5

Platform Improvements in OL9

Overview

Both conversion tracks benefit from a consolidated set of platform improvements that address issues seen during prior Oracle Linux conversions. These apply to **silo-migrate** and to the collector **recovery-image** flow.

This chapter covers the following topics:

Error Reporting and Observability

Standardized error codes and messages across migration tooling, recovery scripts, and the coordinator service. Concise failure reasons surface both on the Database Server and Data Engine side (tables and logs) and in the Skylar One user interface where applicable.

Readiness-Check Transparency

Readiness check failures (RAM, disk, unsupported platform) are reported explicitly and actionably in logs and the Skylar One user interface. Each condition is classified as a hard failure, a warning, or a tunable policy.

Staging Robustness and Cleanup

Retry and backoff handling for image staging tolerates transient network conditions. Stuck staging states are detected and recoverable. Stale entries in **appliance_image_status** and **appliance_image_staging** are cleaned up safely.

Service Life Cycle and Cache Management

The coordinator service stays disabled by default. Misconfiguration is detected and surfaced. Caches and temporary files are cleaned up automatically after successful conversions.

Post-Conversion State Refresh

Appliance OS version and platform state are refreshed promptly after conversion, so OL9 state shows up in the Skylar One user interface and internal tables without lag or manual intervention.

Other Benefits of OL9 Conversion

1. **Node replacement, not migration.** The OL9 node joins the existing cluster as a secondary. No five-stage migration. No standalone outcome that must be re-clustered.
2. **Customer-controlled pace.** Run mixed OL8 and OL9 for a day or two months. Take MRs while mixed. Failover and continue at the customer's pace.
3. **Clean rollback path.** Validation runs while OL8 nodes are still in the cluster. Rollback to OL8 active is preserved if anything is off.

4. **Faster DRBD data seeding.** DRBD enhancements in the OL9 stack should materially reduce seeding time.
5. **Better node-replacement experience for any reason.** Quality-of-life work from the OL9 effort (CRM template updates, config file sync, cleaner cluster join) applies to hardware failure recovery, planned refresh, and VM migration too.
6. **Standardized error reporting.** Concise, actionable failure reasons in logs and user interface, not generic non-zero exit codes.
7. **Transparent readiness checks.** RAM, disk, and platform incompatibility flagged up front, classified as hard failures, warnings, or tunable policies.
8. **Robust image staging.** Retry and backoff handling for transient network conditions. Stuck states are detected and recoverable.
9. **Improved collector conversion.** Better progress feedback and operator interaction in the recovery-image flow.
10. **Stability and scale at production volumes.**

Chapter

6

Support Boundary

Overview

The OL8 to OL9 conversion is a platform life cycle event, delivered through ScienceLogic-controlled tooling and images. The supported model is:

- Skylar One delivers the runtime, including the OL9 base, as a controlled platform component with a tested dependency and update life cycle.
- Customer-managed host responsibilities (domain join, CAC/PIV, host scanning, EDR, host-based access controls) live on the surrounding host environment, not inside the runtime.
- Conversion tooling is validated against the ScienceLogic-supplied OL8 source state and the ScienceLogic-supplied OL9 target state. Conversions of modified runtimes are out of scope.

> **NOTE:** Customer-managed OS controls live on the host side. The Skylar One layer ships as an isolated, supportable runtime. The OL8 to OL9 conversion preserves that boundary on both sides of the transition.

Chapter 7

Frequently Asked Questions (FAQs)

Overview

This chapter contains a list of common questions and their answers related to the Oracle Linux 9 (OL9) conversion process.

This chapter covers the following topics:

General FAQs

Why should I convert to OL9?

Oracle Linux 9 provides updated security baselines, newer kernel features, improved container support, and extended vendor support timelines. For Skylar Onespecifically, OL9 is required for future platform releases and ensures continued ScienceLogic support.

How long will the conversion to OL9 take?

Conversion time varies by topology. Single-node replacements (AIO, standalone DB) typically complete within a maintenance window of a few hours. Mixed-cluster rolling replacements for HA/DR topologies can be staged to minimize downtime. Collector conversions are typically faster, as they run in parallel via the coordinator service.

Will converting to OL9 affect my collectors?

Collector conversion is a separate track from Core CDB conversion. Core CDB and database appliances convert first through silo-migrate. Collectors convert afterward through the recovery-image flow initiated from the Skylar One user interface.

Are mixed OL8 and OL9 clusters supported?

Yes. Corosync and Pacemaker compatibility between OL8 and OL9 allows OL9 nodes to join existing OL8 clusters. This enables rolling in-cluster replacement for HA, DR, and HA+DR topologies. This is a key difference from the OL7-to-OL8 model.

Pre-Conversion Checks

What hardware is required for OL9?

OL9 system requirements are available at https://docs.oracle.com/en/operating-systems/oracle-linux/9/. Verify hardware certification at https://linux.oracle.com/hardware-certifications. In most cases, existing OL8-qualified hardware will support OL9.

What disk space is required?

Use the *capacity check commands* to verify sufficient free logical extents in **em7vg** before starting the Core CDB conversion. For collectors, the recovery-image flow requires enough capacity to hold the OL9 installer and kickstart materials. Insufficient resources will be reported in the Skylar One user interface before conversion begins.

The Conversion Process

Can I convert to OL9 on my own?

The conversion is driven by ScienceLogic-controlled tooling (**silo-migrate** for Core CDB/DB appliances, and the coordinator service plus Skylar One user interface for collectors). Field teams and customers can initiate and monitor the conversion using the procedures in this guide.

What is the preferred conversion order?

Convert Core CDB and database appliances first, then convert collectors. Both tracks can be planned independently, but the collector conversion depends on the DB and DE side being available to run the coordinator service.

What if a collector conversion gets stuck?

See the Troubleshooting section for steps to check service status, review logs, verify image staging, and force a restage if needed.

Security FAQs

Is Skylar One running OL9 securely?

Yes. OL9 includes updated security baselines and is validated against ScienceLogic's STIG and security requirements as part of the platform release process. *Post-conversion validation* includes OL9 security and STIG baseline checks.

Does OL9 support FIPS?

FIPS support on OL9 follows Oracle Linux platform guidelines. Contact ScienceLogic Support for specific FIPS configuration requirements in your environment.

For Additional Questions

I have additional questions. How do I get further help?

If you have additional questions not covered in this guide, contact ScienceLogic Support or your ScienceLogic account team.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010