



Apache Solr



ZooKeeper

TECHNICAL BRIEF

SOLR-18298 / PR #4774 — Session-expiration-aware ZooKeeper recovery

Submitted by: Shrey Narayan

Organization: NextBricks

Reviewer request: AtomicBoolean in ZkController and ZkStateReader

Commit: 21cb72bc750de09961f5f8efee711a5ad73a9cc0

Branch: fix/solr-18298-session-expiry-reconnect

Status: Pushed; focused local tests pass

Prepared: August 26, 2026

Outcome The shared OnDisconnect and OnReconnect adapters keep their general-purpose semantics. ZkController and standalone ZkStateReader now each own an AtomicBoolean that gates full recovery on a preceding LOST event.

Executive assessment

The revised patch follows the reviewer's requested ownership model and removes the original shared WeakHashMap state. This is the cleaner concurrency boundary: session-loss state belongs to the component whose recovery it controls, not to a generic listener interface used by other callers.

- SUSPENDED → RECONNECTED: connection returns, but full SolrCloud session-expiration recovery is skipped.
- LOST → RECONNECTED: the owning component performs recovery exactly once.
- An additional RECONNECTED notification cannot repeat the recovery because compareAndSet atomically consumes the flag.

State-transition contract

Curator sequence	Atomic state	Component behavior
SUSPENDED → RECONNECTED	false → false	No full session-expiration recovery
LOST	false → true	Record loss; ZkController performs existing expiry teardown
LOST → RECONNECTED	true → false	Run recovery once and consume the flag
extra RECONNECTED	false → false	No duplicate recovery



Apache Solr



ZooKeeper

Implementation

ZkController

OnDisconnect continues to receive both transient and terminal state. A transient event returns immediately. A lost session sets the local flag before the existing overseer/election teardown. OnReconnect consumes the flag before entering the expensive recovery body.

```
private final AtomicBoolean zkSessionExpired = new AtomicBoolean();

private void onDisconnect(boolean sessionExpired) {
    if (!sessionExpired) return;
    zkSessionExpired.set(true);
    // existing session-expiration teardown
}

private void onReconnect() {
    if (!zkSessionExpired.compareAndSet(true, false)) return;
    // existing full SolrCloud recovery
}
```

ZkStateReader

Standalone ZkStateReader now registers the unchanged OnDisconnect and OnReconnect adapters and owns the same one-shot atomic state. Watcher reconstruction occurs after LOST, but not after a short network suspension.

```
private final AtomicBoolean zkSessionExpired = new AtomicBoolean();
private final OnDisconnect onDisconnect = this::onDisconnect;
private final OnReconnect onReconnect = this::onReconnect;

// LOST sets the flag; RECONNECTED consumes it and refreshes watchers once.
```

Why this concurrency primitive is correct

- **Visibility:** Curator may deliver state transitions on managed callback threads; AtomicBoolean supplies cross-thread visibility.
- **Atomicity:** compareAndSet combines eligibility and reset, so closely spaced reconnect notifications cannot both enter recovery.
- **Isolation:** ZkController and ZkStateReader cannot interfere with one another or with unrelated listener instances.
- **Lifecycle:** the tactical state is localized and can be deleted when persistent watchers remove the rebuild requirement.

Important nuance Curator documents LOST as the state used when it believes the ZooKeeper session expired; RECONNECTED can follow either SUSPENDED or LOST. That distinction is precisely why RECONNECTED alone is insufficient as the recovery trigger.



Apache Solr



ZooKeeper

Local verification

Check	Result	Evidence
./gradlew tidy	PASS	Solr formatter completed successfully.
Listener adapter test	PASS	2 tests; unchanged shared-interface behavior.
ZkController integration test	PASS	1 test; 3-node ensemble failover skipped recovery, injected expiration triggered it once.
git diff --check	PASS	No whitespace errors.
./gradlew check -x test	ENVIRONMENT	Compilation/validation progressed; root RAT rejected pre-existing repository files in this worktree, unrelated to the five-file patch.
Local SolrCloud	PASS	Solr 11.0.0-SNAPSHOT at <code>***.***.***.***.***;</code> embedded ZooKeeper at <code>***.***.***.***.***;</code>

Focused commands

```
./gradlew :solr:solrj-zookeeper:test --tests org.apache.solr.common.cloud.TestOnReconnectSessionExpiry
./gradlew :solr:core:test --tests org.apache.solr.cloud.ZkControllerTest.testReconnectRecoveryRequiresSessionExpiration
./gradlew tidy
git diff --check
```

Runtime evidence

Figure 1. Patched local Solr Admin dashboard at http://*****/solr. The Versions panel identifies Solr 11.0.0-SNAPSHOT and implementation commit 21cb72bc75.

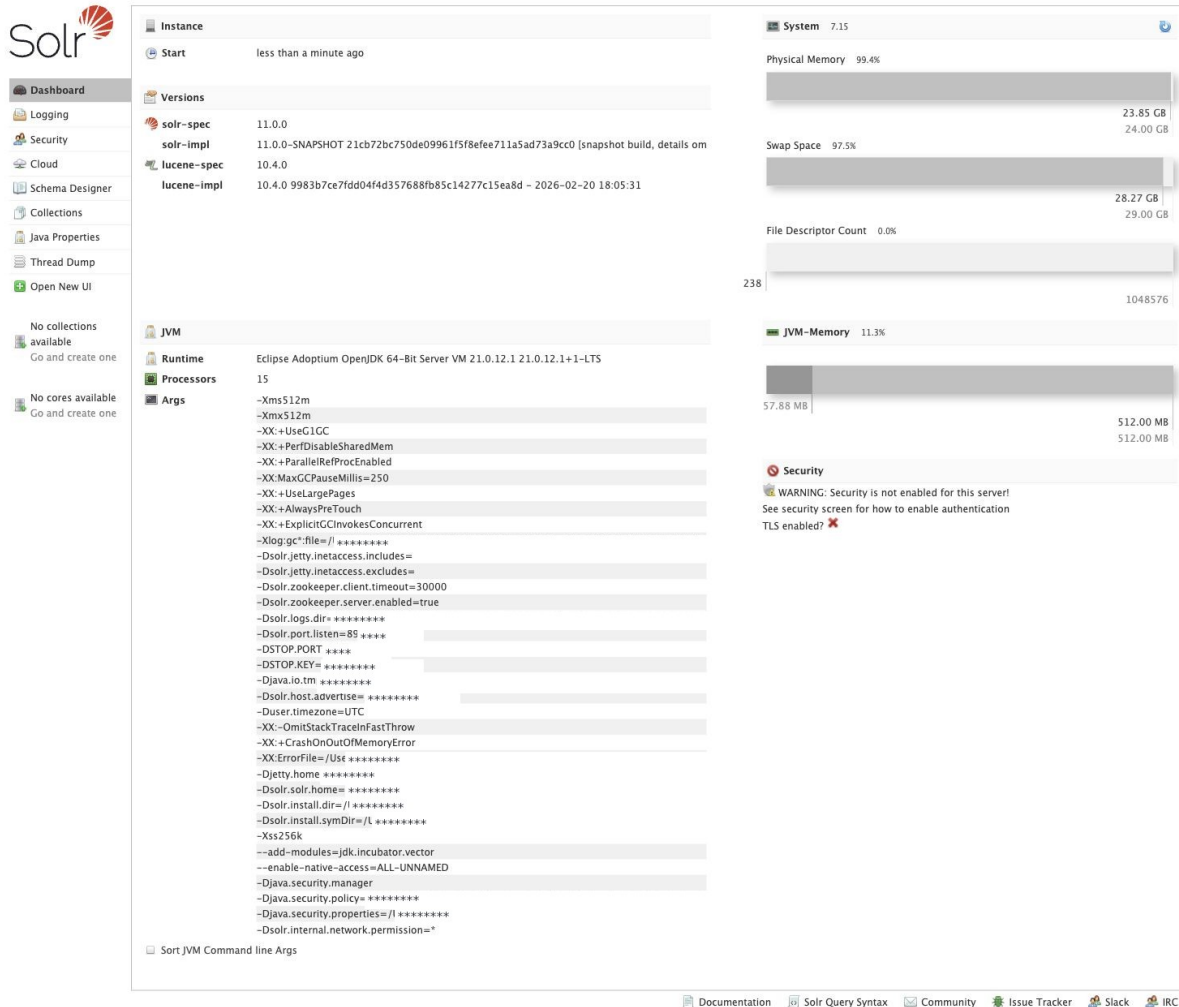


Figure 2. Solr Admin Cloud > ZK Status for the embedded ZooKeeper service. The live status is green, the connection string is `***.***.***.***:***`, and ZooKeeper reports clientPort `***`.

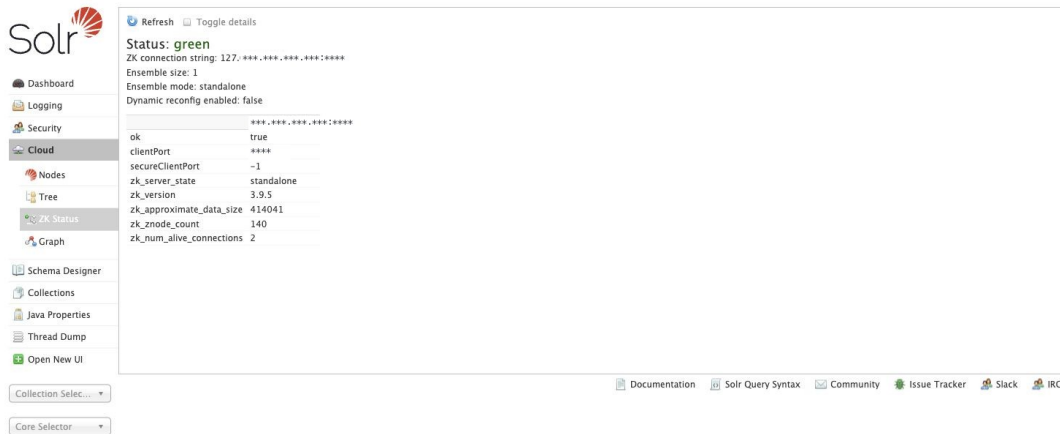


Figure 3. Solr Admin Cloud > Tree with `/live_nodes` expanded to `*****:***_solr` and `/overseer_elect` expanded. These are concrete ZooKeeper paths affected by ZkController session-loss recovery.

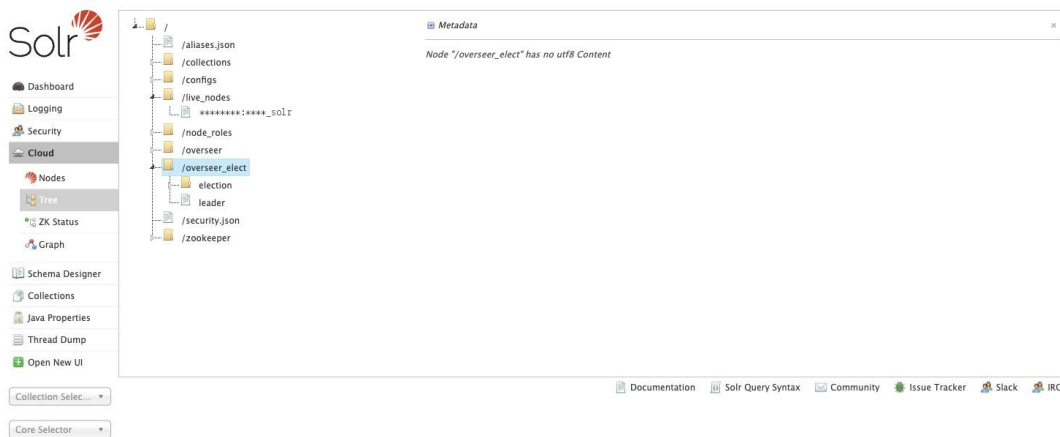


Figure 4. Search Studio configured with the correct Solr HTTP endpoint, `http://*****/solr`, for both source and target. Port `****` is not used as an HTTP Solr URL. The visible HTTP 502 result is expected because the hosted Search Studio backend cannot reach browser-local loopback.

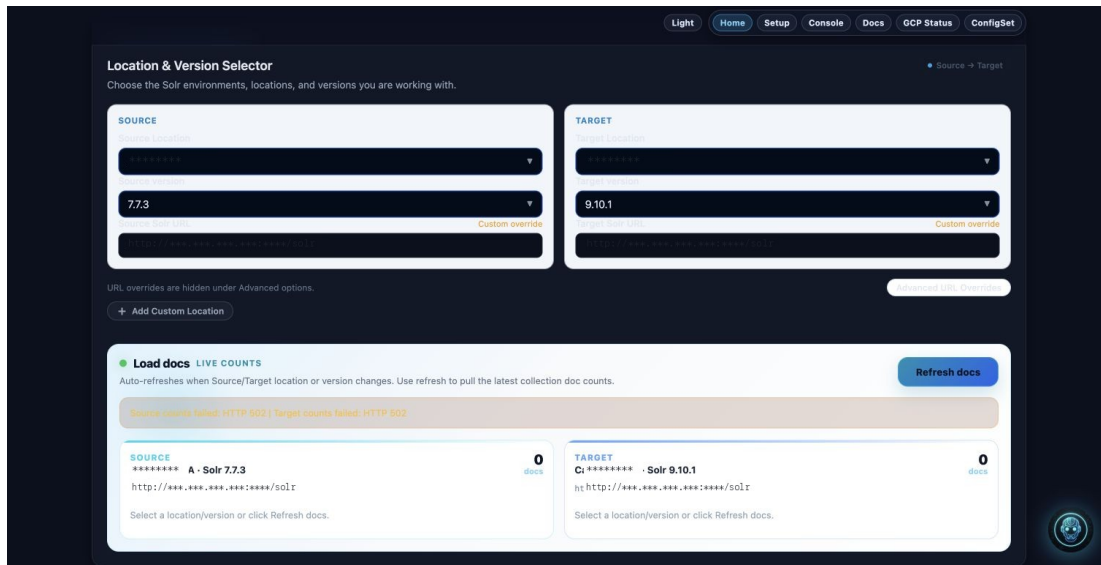
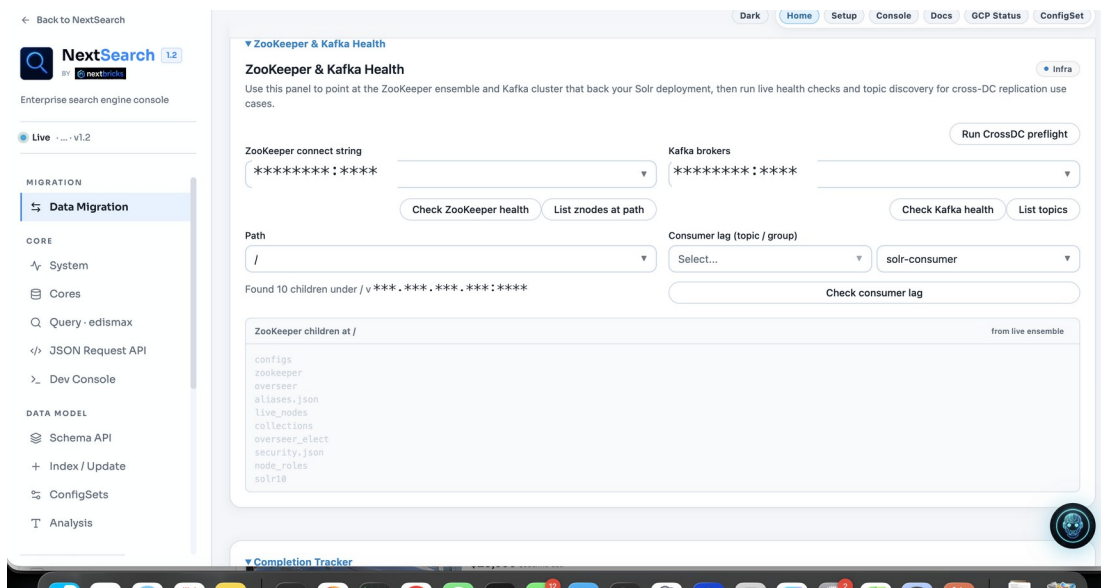


Figure 5. NextSearch 1.2 ZooKeeper & Kafka Health panel connected to the existing ZooKeeper ensemble at *****:****. The response reports 10 children under /, including live_nodes and overseer_elect. This is a separate product-context environment; it is not the patched local SolrCloud instance whose embedded ZooKeeper runs on ***.****.*****.*****.





Documentation basis

Source	Relevance to this fix	Citation
Apache Curator ConnectionState API	Defines SUSPENDED, LOST, and RECONNECTED; LOST is the session-expiration signal used by the patch.	Official documentation
Apache Curator error handling	Explains connection-state notifications and application-specific recovery after session loss.	Official documentation
ZooKeeper Programmer's Guide	Documents sessions, expiration, asynchronous state notifications, and one-time watch semantics.	Official documentation
Solr Reference Guide: cluster types	Documents ZooKeeper's role in SolrCloud cluster state and coordination.	Official documentation
Solr control script reference	Documents embedded ZooKeeper using the Solr port plus 1000; Solr 8983 therefore maps to ZK 9983.	Official documentation
Java AtomicBoolean API	Defines atomic compare-and-set behavior used to consume the one-shot recovery flag safely.	Official documentation

Files changed

File	Purpose
changelog/unreleased/SOLR-18298-zk-reconnect-session-expiry.yml	Record the behavioral fix in the unreleased changelog.
ZkController.java	Own and consume the session-expiration flag around existing recovery.
ZkStateReader.java	Own equivalent one-shot state for standalone watcher refresh.
TestOnReconnectSessionExpiry.java	Verify the unchanged shared-adapter contracts.
ZkControllerTest.java	Exercise transient ensemble failover versus injected session expiration.



Apache Solr



ZooKeeper

Paste-ready reviewer reply

Thanks, Houston — I reworked this along the ownership boundary you suggested. I reverted the semantic changes in `OnDisconnect` and `OnReconnect`, so those shared adapters retain their existing behavior. `ZkController` and standalone `ZkStateReader` now each maintain a private `AtomicBoolean` that is set only when `OnDisconnect` reports `sessionExpired=true` (Curator LOST). Their reconnect handlers use `compareAndSet(true, false)`, so a transient `SUSPENDED → RECONNECTED` does not run full recovery, while `LOST → RECONNECTED` runs it once and atomically consumes the flag.

I added coverage for the unchanged listener contracts and a three-node ZooKeeper integration test that distinguishes ordinary ensemble failover from injected session expiration. `./gradlew tidy`, the two focused test targets (2 listener tests + 1 integration test), and `git diff --check` pass. I also launched the patched build locally as `SolrCloud` on a redacted local port with embedded ZooKeeper on a redacted local port and verified commit `21cb72bc75` in the Admin UI. This keeps the state local and temporary, so it can be removed cleanly when persistent watchers eliminate this recovery path.