Rigel: config - Task #27814

Add runtime cache control functionality to Rigel client

31/05/2025 10:28 PM - Sachin Divekar

Status:	Testing	**Start date:**	31/05/2025
Priority:	Normal	**Due date:**	
Assignee:	Sachin Divekar	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Implement the ability to enable/disable caching at runtime in the Rigel configuration client.

Currently, the Rigel client always uses caching for configuration values. There are legitimate use cases that require runtime cache control:

- **Debugging**: Temporarily disable cache to get fresh values when investigating configuration issues
- **Health checks**: Verify actual etcd connectivity rather than cached responses
- **Critical operations**: Ensure latest values for time-sensitive operations (payment limits, feature flags)
- **Performance testing**: Compare cached vs non-cached performance

The implementation must be thread safe.

Associated revisions

Revision d29a99ba - 31/05/2025 10:57 PM - Sachin Divekar

Implement cache control #27814

Implement runtime cache control with thread-safe atomic operations

Changes:

- Add private cacheEnabled field using atomic.Bool for lock-free concurrent access
- Add EnableCaching()/DisableCaching() methods to control cache behavior at runtime
- Add IsCachingEnabled() to check current caching state
- Update Get() method to respect caching state - bypasses cache when disabled
- Update WatchConfig() to respect caching state when processing events
- Maintain existing WatchConfig behavior - only updates existing cache keys
- Update all constructors to initialize caching as enabled by default

Revision df11f3e9 - 01/06/2025 04:02 PM - Sachin Divekar

Implement schema caching #27814

- Add schema caching in Rigel client to avoid repeated etcd calls for schema validation
- Schema fields are now cached in memory when caching is enabled
- DisableCaching() properly clears schema cache to ensure fresh lookups
- Add InvalidateSchemaCache() method for manual cache invalidation
- Add examples to demonstrate the features

History

#1 - 31/05/2025 10:28 PM - Sachin Divekar

- Status changed from New to In Progress

#2 - 31/05/2025 10:28 PM - Sachin Divekar

- % Done changed from 0 to 100

#3 - 31/05/2025 10:30 PM - Sachin Divekar

- Status changed from In Progress to Testing

Use Case Examples:

```
// Debugging production issue
```

```
client.DisableCaching()
freshValue := client.Get("problematic_key")
client.EnableCaching()

// Health check
func healthCheck() error {
    client.DisableCaching()
    defer client.EnableCaching()
    _, err := client.Get("health_key")
    return err
}

// Critical operation
func processPayment() {
    client.DisableCaching()
    defer client.EnableCaching()
    limit := client.GetFloat("payment_limit") // Always fresh
}
```