

Commvault Analytics App

Enhancement Plan

Date: 2025-11-14 **Focus:** Advanced Retention & Pruning Problem Identification

Executive Summary

Based on deep research into Commvault-specific aging and pruning issues, this plan outlines comprehensive enhancements to transform the app into a powerful diagnostic and analytics tool for identifying and resolving retention problems.

Core Problem Areas Identified

1. **Two-Stage Failure Pattern:** Aging (logical) delays + Pruning (physical) failures
 2. **Deduplication Complexity:** Reference counting, pending deletes, DDB sealing
 3. **Visibility Gaps:** No real-time pruning metrics, limited DDB health monitoring
 4. **Retention Misconfiguration:** Cycle requirements causing 50-100% longer retention
 5. **Silent Failures:** Pruning can fail without obvious job errors
-

Research Findings Summary

Critical Commvault Concepts

1. Aging vs Pruning (The Two-Stage Process)

- **Aging (Logical):** Marks jobs as eligible for deletion, updates metadata only, NO space freed
- **Pruning (Physical):** Actually deletes data blocks from disk, DOES free space
- **Key Insight:** Aging can succeed while pruning fails silently

2. Retention Logic (AND Operation)

```
Data Eligible for Aging = (Days_Passed > Retention_Days)  
AND  
(Cycles_Complete > Retention_Cycles)
```

- **Default Commvault Setting:** 15 days + 2 cycles
- **Problem:** With weekly full backups, 2 cycles = 14 days minimum
- **Result:** 15-day retention actually keeps data for ~21 days (40% longer)

3. Deduplication Pruning Process

1. Job aged → Reference counters decremented for each block
2. Block ref count reaches 0 → Added to MMDeletedAF table
3. Mark and Sweep operation → Identifies pruneable chunks
4. Physical pruning → Deletes chunks from disk (logged in SIDBPhysicalDeletes.log)
5. Space freed → DDB statistics updated

4. Micropruning vs Macro Pruning

- **Micropruning:** Delete individual blocks as jobs age (enabled by default for cloud)
- **Macro Pruning:** Seal DDB, wait for all jobs to age, delete entire DDB
- **WORM Storage:** MUST use macro pruning (3x storage requirement)
- **Best Practice:** Use micropruning except for WORM/archive cloud

5. Common Failure Points

Aging Failures: - Cycle requirements not met (backup failures prevent cycle completion)
- Infinite retention (-1/-1 configuration) - Aging disabled (enableDataAging = 0) -
Storage policy changes without full backup - Insufficient permissions (Error 32:465)

Pruning Failures: - MediaAgent offline or overloaded - DDB sealed/corrupted - Mount paths offline or inaccessible - High pending deletes queue (>100,000 critical) - Pruning operation windows blocking execution - Resource exhaustion (CPU/Memory/Disk I/O) - Managed Disk Space enabled (delays pruning until watermark)

6. Key Log Files & Monitoring

- **SIDBPrune.log:** Pruning activity on MediaAgent
- **SIDBPhysicalDeletes.log:** Physical deletion operations
- **SIDBEngine.log:** Mark and Sweep operations, pending delete counts
- **CVMA.log:** MediaAgent operations (non-dedup)
- **MMDeletedAF table:** Database table of aged data awaiting pruning
- **Mark and Sweep:** Should run daily (check SIDBEngine.log)

7. Retention Codes (Why Data Isn't Aging)

- **BASIC_CYCLES** - Cycle requirements not met
- **BASIC_DAYS** - Day requirements not met
- **NOT_COPIED** - Job awaiting auxiliary copy
- **INFINITE** - Infinite retention configured
- **JOBRUNNING** - Job currently executing
- **EXTENDED_RETENTION** - Extended retention rule active

8. REST API Endpoints

- **DDB Details:** `GET /DDBInformation/{ddbStoreId}` - Returns pending delete counts, statistics
- **Job Details:** `GET /Job/{jobId}` - Job information

- **Storage Policies:** Various endpoints for policy details
 - **Note:** No dedicated API for pruning statistics (must query database or parse logs)
-

Phase 1: Dashboard & Overview Analytics

1.1 Retention Health Dashboard

Location: New route [/dashboard/retention](#)

Metrics to Display: - Total retention rules analyzed - Rules with aging disabled (CRITICAL) - Rules with infinite retention (INFO) - Rules with inefficient short-term policies (HIGH PRIORITY) - Rules with high cycle requirements (MEDIUM) - Rules with optimal configuration (OK)

Visualizations: - Pie chart: Retention issue severity breakdown - Bar chart: Top 20 plans with most issues - Trend line: Aging health over time (if historical data available) - Heat map: Storage policies by issue type

Status Indicators: - ● CRITICAL: Aging disabled (0% space reclamation) - ● HIGH: Inefficient short-term (≤ 30 days + 2+ cycles) - ● MEDIUM: High cycles (3+ cycles) - ● INFO: Infinite retention - ● OK: Optimal configuration

1.2 Storage Pool Health Dashboard

Location: New route [/dashboard/storage](#)

Metrics to Display: - Total storage pools - Pools by status (Critical <10%, Warning 10-20%, Low 20-30%, OK >30%) - Total capacity vs free space - Top 10 pools by % full - Pools requiring immediate action

Visualizations: - Gauge charts: Per-pool % free - Stacked bar: Capacity vs used vs free - Alert list: Pools needing emergency pruning - Timeline: Pool fill rate (GB/day) if historical data available

Actionable Insights: - "3 pools critically low - Manual pruning recommended" - "Apex GDP at 1.97% free - IMMEDIATE ACTION REQUIRED" - "11 pools below 20% - Evidence of pruning failure"

1.3 Pruning Health Dashboard

Location: New route </dashboard/pruning>

Metrics to Display (if data available): - MediaAgents online vs offline - DDB status (Active vs Sealed) - Pending delete counts per DDB - Last successful pruning operation per pool - Pruning job success rate (last 30 days)

Key Indicators: - MediaAgent health: "243 MediaAgents - X online, Y offline" - DDB health: "X Active, Y Sealed (waiting for aging)" - Pending deletes: "DDB1: 45,234 pending (WARNING)" - Pruning trend: "Last 7 days: 85% success rate"

Phase 2: Advanced Analytics & Problem Identification

2.1 Cycle Completion Analyzer

Purpose: Identify plans where cycle requirements are blocking aging

Analysis: 1. Query jobs table for full backup frequency per plan 2. Calculate average cycle duration (days between full backups) 3. Compare cycle duration vs retention requirements 4. Flag plans where cycle requirements extend retention by >30%

Output: - Table: Plan | Retention Days | Retention Cycles | Avg Cycle Duration | Effective Retention | Delta - Example: "Apex Plan | 14 days | 2 cycles | 7 days/cycle | 21 days | +50% delay" - Recommendations: "Reduce to 1 cycle to achieve true 14-day retention"

SQL Query Example:

```

WITH full_backup_intervals AS (
    SELECT
        planName,
        clientName,
        JULIANDAY(startTime) - LAG(JULIANDAY(startTime)) OVER (
            PARTITION BY planName, clientName ORDER BY startTime
        ) as days_between_fulls
    FROM jobs
    WHERE jobType LIKE '%Full%'
)
SELECT
    p.planName,
    rr.retainBackupDataForDays,
    rr.retainBackupDataForCycles,
    AVG(fbi.days_between_fulls) as avg_cycle_days,
    MAX(rr.retainBackupDataForDays, rr.retainBackupDataForCycles * AVG(fbi.days_between_fulls))
FROM plans p
JOIN retention_rules rr ON p.planId = rr.parentId
LEFT JOIN full_backup_intervals fbi ON p.planName = fbi.planName
GROUP BY p.planName

```

2.2 Storage Reclamation Forecaster

Purpose: Predict space to be reclaimed if aging/pruning fixes are applied

Analysis: 1. Identify all aged jobs (or jobs eligible for aging with fixes) 2. Sum data size for aged jobs per storage pool 3. For deduplication: Estimate unique vs shared blocks 4. Calculate potential space recovery

Output: - Table: Storage Pool | Current Free | Aged Data Size | Potential Free Space | % Improvement - Example: "Apex GDP | 1.97% (50 GB) | 2.1 TB aged data | 45% free after pruning" - Timeline: "Expected recovery: Day 1: +5%, Week 1: +20%, Month 1: +45%"

Caveats: - Deduplication reduces actual space freed (shared blocks) - Display warning: "Estimates for dedup storage are approximate"

2.3 Backup Job Success Analyzer

Purpose: Identify failed backups preventing cycle completion

Analysis: 1. Query jobs for failed full backups in last 30 days 2. Group by plan/client 3. Calculate full backup success rate per plan 4. Flag plans with <95% full backup success rate

Output: - Table: Plan | Client | Last Full Backup | Status | Success Rate (30d) | Impact - Example: "Apex Plan | Server123 | 2025-10-15 (30 days ago) | Failed | 60% |  Blocking cycle 2" - Alert: "5 plans have failing full backups - Cycles cannot complete, aging blocked"

Recommendations: - "Fix backup failures before addressing retention policies" - "Plans with failed backups will NOT benefit from cycle reduction"

2.4 Retention Policy Optimizer

Purpose: Recommend optimal retention configurations

Analysis: 1. Identify all retention rules 2. Categorize by intent: Short-term (<30 days), Standard (30-90), Long-term (>90) 3. Compare against best practices 4. Generate recommendations

Best Practice Rules: - **Short-term (<30 days):** 1 cycle only - **Standard (30-90 days):** 1-2 cycles - **Long-term (>90 days):** 2 cycles - **Disaster Recovery:** 2-3 cycles minimum - **Compliance/Archive:** Days-based only (0 cycles) or extended retention

Output: - Table: Plan | Current Config | Recommended Config | Rationale | Impact - Example: "Apex Plan | 14d + 2c | 14d + 1c | Short-term optimization | -7 days aging delay" - Bulk actions: "Apply optimal config to 130 plans in one click" (future enhancement)

2.5 Deduplication Impact Analyzer

Purpose: Understand how deduplication affects space reclamation

Analysis: 1. Identify deduplicated storage policies/pools 2. Calculate average deduplication ratio 3. Estimate shared block percentage 4. Warn about pruning complexity

Output: - "75 storage pools use deduplication (95% of environment)" - "Average dedup ratio: 8.5:1 (excellent)" - "WARNING: Pruning requires ALL jobs sharing blocks to age before space is freed" - "Recommendation: Review extended retention rules - they delay pruning for all jobs"

Key Insight: - "A single job with long retention can prevent pruning of hundreds of other aged jobs" - "This is why extended retention on dedup storage is problematic"

Phase 3: Real-Time Monitoring & Alerting

3.1 Job-Based Monitoring

Data Collection: - Poll jobs API every 5 minutes for new aging/pruning jobs - Store job results in database with timestamp - Track: Job ID, Type (Aging/Pruning), Status, Start/End Time, Data Aged/Pruned

Alerts: - "Data Aging job failed 3 times in last 24 hours" - "No pruning activity detected in 7 days on Storage Pool X" - "Aging job completed but 0 jobs aged - Check retention rules"

3.2 Storage Pool Monitoring

Data Collection: - Poll storage pools API every hour - Track % free over time - Calculate fill rate (GB/day)

Alerts: - "Apex GDP filling at 50 GB/day - Will reach 0% in 2 days" - "3 pools crossed 10% threshold - Emergency pruning needed" - "Southern_Sun_Durban has been <5% free for 14 days - Pruning failure confirmed"

3.3 MediaAgent Health Monitoring

Data Collection: - Poll MediaAgents API every 15 minutes - Track online/offline status - Alert on status changes

Alerts: - "MediaAgent CVHSMAN01 went offline - Pruning blocked for 5 pools" - "3 MediaAgents offline for >24 hours - Check connectivity"

3.4 Retention Rule Drift Detection

Data Collection: - Snapshot retention rules daily - Compare to previous snapshot - Track changes

Alerts: - "Retention rule changed on Apex Plan: 14d+1c → 14d+2c (Optimization regressed!)" - "5 rules changed in last 7 days - Review changes" - "New infinite retention rule added on Google Drive Plan"

Phase 4: Diagnostic Tools & Utilities

4.1 Retention Rule Checker

Location: </tools/retention-checker>

Functionality: - Upload retention rule config OR select plan from dropdown - Instant analysis: - Optimal configuration -  Warnings (inefficient cycles, high retention) -  Errors (aging disabled, infinite retention) - Recommendations with before/after comparison - "Try It" simulator: "If we change to 1 cycle, aging will occur X days faster"

4.2 Storage Pool Health Predictor

Location: </tools/storage-predictor>

Functionality: - Select storage pool - Input: Current fill rate (GB/day) or auto-calculate from history - Output: - "Days until pool is full: 15" - "Recommended action: Manual pruning + Fix retention on 3 plans feeding this pool" - Timeline chart showing projected fill

4.3 Plan Impact Analyzer

Location: `/tools/plan-impact`

Functionality: - Select plan - Show all affected resources: - Clients using this plan - Storage policies used - Storage pools used - Retention rules - Current aging status - Jobs awaiting aging - Impact simulation: "If we change retention to 1 cycle:" - "20 jobs will age immediately" - "Estimated space freed: 500 GB" - "3 storage pools will improve by 5%"

4.4 Backup Schedule Analyzer

Location: `/tools/schedule-analyzer`

Functionality: - Analyze backup schedules vs retention requirements - Identify plans where backup frequency doesn't support retention - Example issues: - "Plan requires 2 cycles but full backups run monthly (60-day effective retention)" - "Synthetic full configured but not running - Cycles never complete" - Recommendations: "Increase full backup frequency to weekly to support 2-cycle retention"

4.5 Bulk Remediation Planner

Location: `/tools/bulk-remediation`

Functionality: - Select issue type (e.g., "Inefficient short-term policies") - Show all affected plans (130 plans) - Generate remediation script: - Option 1: Manual steps (UI navigation) - Option 2: API calls (for automation) - Option 3: XML config export/import - Safety checks: "Backup current config before applying changes" - Rollback plan included

Phase 5: Reporting & Documentation

5.1 Executive Reports

Available Reports:

1. Storage Reclamation Opportunity Report

2. Total aged data not yet pruned
3. Potential space recovery by pool
4. ROI: Cost savings from reclaimed space
5. Timeline for recovery

6. Retention Policy Compliance Report

7. Plans following best practices: X%
8. Plans with issues: Y%
9. Top 10 problem plans
10. Remediation roadmap

11. Backup Infrastructure Health Report

12. MediaAgent status summary
13. DDB health summary
14. Pruning success rate
15. Aging success rate
16. Areas of concern

17. Trend Analysis Report

18. Storage growth rate
19. Aging effectiveness over time
20. Pruning effectiveness over time
21. Issue resolution progress

5.2 Technical Deep-Dive Reports

Available Reports:

1. **Cycle Completion Analysis Report**
2. Per-plan cycle duration analysis
3. Full backup frequency analysis
4. Recommendations for cycle optimization

5. Deduplication Efficiency Report

6. Dedup ratios per pool
7. Shared block analysis
8. Pruning complexity assessment
9. Extended retention impact

10. Failed Backup Investigation Report

11. All failed full backups (30 days)
12. Impact on cycle completion
13. Root cause categories
14. Remediation priorities

5.3 Scheduled Reports

Automation: - Daily: Storage pool status email (if any pools <20%) - Weekly: Retention health summary - Monthly: Comprehensive environment analysis - Quarterly: Trend analysis with recommendations

Phase 6: API & Data Enhancements

6.1 Additional Data Collection

New API Endpoints to Fetch:

1. **DDB Information** (`GET /DDBInformation/{ddbStoreId}`)

2. Pending delete counts
3. Reference count statistics
4. DDB status (Active/Sealed)

5. Last verification date

6. **Job Details Enhanced** (`GET /Job/{jobId}`)

7. Fetch aging/pruning jobs specifically
8. Extract aged job counts
9. Extract pruned data size
10. Error details for failed jobs

11. **Schedule Information**

12. Backup schedule details per plan
13. Full backup frequency
14. Synthetic full configuration

15. **Extended Retention Rules**

16. First/second extended retention details
17. Apply to which jobs

6.2 Database Schema Enhancements

New Tables:

1. `job_history_detailed`
2. Store ALL job details (not just last 100)
3. Full backup jobs flagged
4. Success/failure tracking

5. Enable cycle duration calculations
6. **aging_pruning_jobs**
7. Dedicated table for aging/pruning operations
8. Track jobs aged per run
9. Track space freed per run
10. Performance metrics
11. **storage_pool_history**
12. Snapshot storage pool stats every hour
13. Enable trend analysis
14. Calculate fill rates
15. Detect pruning effectiveness
16. **retention_rule_history**
17. Track retention rule changes over time
18. Audit trail
19. Drift detection
20. **mediaagent_health**
21. MediaAgent status snapshots
22. Uptime tracking
23. Offline duration tracking
24. Correlation with pruning failures
25. **ddb_statistics**
26. DDB stats over time
27. Pending delete trends

28. Reference count trends

29. Seal/verification history

6.3 Data Analysis Engine

Background Processing:

1. Nightly Analysis Job

2. Run all analytical queries

3. Pre-calculate metrics

4. Store in summary tables

5. Enable fast dashboard loading

6. Anomaly Detection

7. Compare current metrics to historical baseline

8. Flag unusual patterns:

- Storage filling faster than normal

- Aging jobs taking longer than usual

- Sudden spike in failed backups

9. Generate automatic alerts

10. Predictive Modeling

11. Machine learning model for storage fill prediction

12. Predict aging failures based on job patterns

13. Recommend optimal retention based on actual usage

Phase 7: User Experience Enhancements

7.1 Guided Troubleshooting Wizard

Location: /tools/troubleshoot

Flow: 1. **Select Problem Type:** - "Storage pools are full" - "Aging jobs are failing" - "Data is not aging fast enough" - "Pruning is not freeing space"

1. **Automated Diagnosis:**

2. Run checks based on problem type
3. Display findings in order of likelihood
4. Example for "Storage pools full": Aging is enabled (PASS) ⚠️ 130 rules have inefficient cycles (LIKELY CAUSE) ✗ 11 pools <20% free (SYMPTOM) ⚠️ 243 MediaAgents not verified (CHECK)

5. **Step-by-Step Remediation:**

6. Prioritized action plan
7. Links to relevant configuration screens
8. Copy-paste API calls for automation
9. Expected outcome for each step

10. **Verification:**

11. Re-run checks after remediation
12. Confirm issues resolved
13. Monitor for 7 days to ensure sustained fix

7.2 Interactive Visualizations

Technologies: Chart.js, D3.js, or Plotly

Visualizations:

1. **Sankey Diagram: Data Flow**

2. Plans → Storage Policies → Storage Pools

3. Width = data volume

4. Color = health status

5. Click to drill down

6. Network Graph: Dependencies

7. Show relationships between plans, policies, pools, MediaAgents

8. Identify single points of failure

9. Highlight offline MediaAgents blocking multiple pools

10. Timeline: Aging & Pruning Events

11. Horizontal timeline of aging jobs

12. Overlay pruning jobs

13. Identify gaps (pruning not running)

14. Compare to storage pool fill rate

15. Heat Map: Plan vs Issue Type

16. Rows: Plans

17. Columns: Issue types

18. Color intensity: Severity

19. Quickly identify "problem plans"

7.3 Search & Filter Capabilities

Global Search: - Search for: Plans, Clients, Storage Policies, Jobs, Storage Pools - Results: Show relevant details + health status - Quick actions: "View Details", "Analyze Retention", "Check Health"

Advanced Filters: - Filter plans by: Issue type, Storage policy, Retention days/cycles, Status - Filter storage pools by: % free, MediaAgent, Type, Status - Filter jobs by: Type,

Status, Date range, Plan, Client - Save filters as presets: "Show critical issues only"

7.4 Contextual Help & Documentation

Features: - Hover tooltips explaining technical terms - "?" icon next to each metric with detailed explanation - Link to official Commvault documentation - Embedded tutorial videos (if available) - FAQ section based on research findings

7.5 Mobile-Responsive Design

Priorities: - Dashboard views optimized for mobile - Critical alerts accessible on mobile - Key metrics available on phone - Responsive tables (horizontal scroll or card view)

Phase 8: Integration & Automation

8.1 Integration with Commvault

Potential Integrations:

1. **Direct Remediation** (if API supports write operations):
 2. Update retention rules from app
 3. Trigger manual aging/pruning jobs
 4. Modify storage pool settings
5. Requires: User permissions, API authentication
6. **CommCell Console Link**
 7. Deep links to specific configs in CommCell Console
 8. Example: "Edit this retention rule" → Opens CommCell Console to exact location
9. **API-Based Automation**

10. Generate API scripts for bulk changes
11. Example: Python script to update 130 retention rules
12. Validate before execution
13. Log all changes

8.2 Notification Integrations

Supported Channels: - Email: Scheduled reports, critical alerts - Slack/Teams: Real-time alerts, daily summaries - SNMP Traps: For SIEM integration - Webhooks: Custom integrations - SMS: Emergency alerts (storage critically full)

8.3 Export & Import

Export Formats: - CSV: Tabular data - JSON: API-ready format - Excel: Multi-sheet reports with formatting - PDF: Executive reports

Import Capabilities: - Import retention rules for analysis - Import historical data for trending - Bulk configuration changes via CSV/JSON

Implementation Roadmap

Sprint 1 (Week 1-2): Foundation

- Enhanced data collection (DDB stats, job details, schedules)
- Database schema updates
- Basic retention health dashboard
- Storage pool health dashboard

Sprint 2 (Week 3-4): Core Analytics

- Cycle completion analyzer
- Storage reclamation forecaster
- Backup job success analyzer
- Retention policy optimizer

Sprint 3 (Week 5-6): Monitoring

- Job-based monitoring with database storage
- Storage pool trend tracking
- MediaAgent health monitoring
- Basic alerting system

Sprint 4 (Week 7-8): Diagnostic Tools

- Retention rule checker
- Storage pool health predictor
- Plan impact analyzer
- Guided troubleshooting wizard

Sprint 5 (Week 9-10): Reporting & UX

- Executive reports
- Technical deep-dive reports
- Interactive visualizations
- Search and filter capabilities

Sprint 6 (Week 11-12): Integration & Polish

- Notification integrations
- Export/import capabilities

- Mobile-responsive design
 - Documentation and help system
 - Testing and optimization
-

Key Metrics for Success

Technical Metrics:

- **Retention Health Score:** % of rules following best practices (Target: >90%)
- **Storage Reclamation Efficiency:** GB freed per aging cycle (Track trend)
- **Pruning Success Rate:** % of aged data successfully pruned (Target: >95%)
- **Aging Cycle Efficiency:** Actual vs intended retention (Target: <10% delta)
- **Time to Diagnose:** Minutes to identify root cause (Target: <5 min)

Business Metrics:

- **Storage Cost Savings:** \$ saved from reclaimed space
 - **Manual Intervention Reduction:** Hours saved per week
 - **Issue Detection Time:** Hours to detect pruning failure (Target: <24 hours)
 - **Issue Resolution Time:** Days to resolve aging issues (Target: <7 days)
 - **User Adoption:** % of admins using app daily (Target: >80%)
-

Risk Mitigation

Technical Risks:

- 1. API Limitations**
2. Risk: Some data not available via API
3. Mitigation: Parse log files, query database directly (read-only), document limitations
- 4. Performance Impact**
5. Risk: Frequent API polling impacts CommServe
6. Mitigation: Configurable polling intervals, intelligent caching, fetch only changed data
- 7. Data Volume**
8. Risk: Large environments = millions of jobs
9. Mitigation: Pagination, date range limits, database indexing, background processing
- 10. Complex Calculations**
11. Risk: Dedup shared block calculations are estimates
12. Mitigation: Clear disclaimers, conservative estimates, validate against actual results

Operational Risks:

- 1. False Positives**
2. Risk: Alert fatigue from incorrect alerts
3. Mitigation: Tunable thresholds, alert suppression, "snooze" functionality
- 4. Configuration Drift**
5. Risk: Commvault changes made outside app
6. Mitigation: Refresh data regularly, show "last updated" timestamps, manual refresh button
- 7. User Error**

8. Risk: Misinterpreting metrics, making wrong changes
 9. Mitigation: Clear documentation, contextual help, confirmation dialogs, audit trail
-

Technical Stack Recommendations

Backend:

- **Framework:** Flask (current) - Keep existing
- **Database:** SQLite (current) for single-user, consider PostgreSQL for multi-user
- **Background Jobs:** APScheduler for periodic data fetching
- **Caching:** Flask-Caching (Redis backend for production)

Frontend:

- **Charting:** Chart.js (simple) or Plotly.js (advanced/interactive)
- **UI Framework:** Bootstrap 5 (current) or Tailwind CSS
- **Icons:** Font Awesome or Bootstrap Icons
- **Interactivity:** Alpine.js (lightweight) or Vue.js (full-featured)

Monitoring & Alerting:

- **Email:** smtplib (Python built-in)
- **Slack:** slack-sdk
- **Database:** Store alerts with status (pending/sent/acknowledged)

Testing:

- **Unit Tests:** pytest
- **API Mocking:** responses or requests-mock

- **Load Testing:** locust (if scaling to multiple users)
-

Security Considerations

1. **API Credentials:** Store encrypted, never in code, use environment variables
 2. **Database Security:** Encrypt sensitive data, use parameterized queries (already doing)
 3. **User Authentication:** Add login system for multi-user deployment
 4. **Role-Based Access:** View-only vs Admin roles
 5. **Audit Logging:** Log all configuration changes, data exports, bulk operations
 6. **HTTPS:** Enforce HTTPS in production
 7. **Input Validation:** Sanitize all user inputs, API responses
 8. **Rate Limiting:** Prevent API abuse, DoS protection
-

Documentation Requirements

1. **User Guide:** How to use each feature, interpret metrics, troubleshoot issues
 2. **Admin Guide:** Installation, configuration, maintenance, backup/restore
 3. **API Documentation:** If exposing APIs for integrations
 4. **Troubleshooting Guide:** Common issues and solutions
 5. **Architecture Documentation:** System design, data flow, component interactions
 6. **Commvault Concepts:** Glossary, aging/pruning explanation, best practices
 7. **Video Tutorials:** Screen recordings for key workflows
-

Conclusion

This enhancement plan transforms the Commvault API app from a simple data viewer into a comprehensive retention and pruning diagnostic platform. By implementing these features, administrators will be able to:

1. **Quickly identify** the root causes of storage reclamation failures
2. **Predict** storage capacity issues before they cause backup failures
3. **Optimize** retention policies for faster space reclamation
4. **Monitor** aging and pruning operations in real-time
5. **Automate** remediation of common retention configuration issues
6. **Prove ROI** through storage cost savings

The phased approach allows for incremental value delivery while building toward a comprehensive solution. Each phase delivers standalone value while laying the groundwork for subsequent phases.

Next Step: Review this plan, prioritize phases based on immediate needs, and begin Sprint 1 implementation.

Report Generated: 2025-11-14 **Based On:** Deep research into Commvault aging/pruning issues, community forums, official documentation **Target Environment:** Commvault environments with 200+ plans, 500+ retention rules, 50+ storage pools