

WAMAS® Lift & Store 2.8

Referenzhandbuch: 20 Schlüssel-Background-Jobs & Cronjob-Automatisierung

Systemarchitektur & Integrationsleitfaden:

In einer modernen Intralogistikumgebung mit WAMAS® Lift & Store 2.8 steuert die Software automatisierte Vertikalliftsysteme (SSI LOGIMAT) in Echtzeit. Zur Aufrechterhaltung der Systemleistung, Datenkonsistenz und Prozesssicherheit sind automatisierte Hintergrundprozesse unerlässlich.

WICHTIGER ARCHITEKTUR-GRUNDSATZ: Auszuführende Skripte dürfen niemals direkte SQL-UPDATE-Befehle auf Kerntabellen (wie STOCK_OBJECT) durchführen. Statusänderungen und Bestandsanpassungen müssen stets als Eingangstelegramme in die WAMAS-Inboxes (z. B. WCS_STOCKATTRIBUTECHANGE oder inSPM) eingespeist werden. Dadurch übernimmt der interne **WAMAS Background Worker** die fachliche Validierung, Auslösen von Event-Triggern, SPS-Synchronisation sowie die lückenlose Protokollierung im Audit-Trail.

JOB 01: Automatische MHD-Bestandssperre (BBD Auto-Block)

Kategorie: Bestandsführung & Haltbarkeit | **Cron:** 5 0 *** (Täglich 00:05) | **WAMAS-Kontext:** MD002, DI002, WCS_STOCKATTRIBUTECHANGE

Identifiziert täglich um Mitternacht alle Bestände im SSI LOGIMAT, deren Mindesthaltbarkeitsdatum (bb_date) abgelaufen ist, und speist ein Attributänderungs-Telegramm ein, um den Status auf BLOCKED zu setzen.

Oracle SQL:

```
INSERT INTO WCS_STOCKATTRIBUTECHANGE (ID, SYS_PARTNER_ID, RECORD_TYPE, CLIENT_ID, ITEM_NO, VARIANT,
BATCH, BB_DATE, STOCK_RATING_FROM, STOCK_RATING_TO, PROCESS_STATE, CREATION_DATE)
SELECT SEQ_WCS_STOCKATTRCHANGE.NEXTVAL, 'CRON_BBD', 'WCSSTOCKATTRIBUTECHANGE00004', CLIENT_ID,
ITEM_NO, VARIANT, BATCH, BB_DATE, 'AVAILABLE', 'BLOCKED', 'NEW', SYSDATE
FROM STOCK_OBJECT WHERE BB_DATE < TRUNC(SYSDATE) AND STOCK_RATING_ID = 'AVAILABLE' AND QTY_AVAILABLE
> 0;
```

MS SQL Server:

```
INSERT INTO WCS_STOCKATTRIBUTECHANGE (SYS_PARTNER_ID, RECORD_TYPE, CLIENT_ID, ITEM_NO, VARIANT,
BATCH, BB_DATE, STOCK_RATING_FROM, STOCK_RATING_TO, PROCESS_STATE, CREATION_DATE)
SELECT 'CRON_BBD', 'WCSSTOCKATTRIBUTECHANGE00004', CLIENT_ID, ITEM_NO, VARIANT, BATCH, BB_DATE,
'AVAILABLE', 'BLOCKED', 'NEW', GETDATE()
FROM STOCK_OBJECT WHERE BB_DATE < CAST(GETDATE() AS DATE) AND STOCK_RATING_ID = 'AVAILABLE' AND
QTY_AVAILABLE > 0;
```

JOB 02: Prüfmittel- & DGUV V3 Kalibrierungs-Vorwarnung (CSIA 1 Alert)

Kategorie: Bestandsführung & Haltbarkeit | **Cron:** 0 6 *** (Täglich 06:00) | **WAMAS-Kontext:** MD002 (CSIA 01), LM119, LM130

Liest das Freifeld CSIA 01 (Prüf- / Kalibrierdatum von Spezialwerkzeugen) aus. Liegt das Datum innerhalb der nächsten 14 Tage, wird eine E-Mail-Warnung an den Werkzeugwart zur Terminplanung gesendet.

Oracle SQL:

```
SELECT s.CLIENT_ID, s.ITEM_NO, s.CSIA1_VALUE AS PRUEFdatum, s.CURRENT_LOCATION_ID
FROM STOCK_OBJECT s WHERE s.CSIA1_VALUE IS NOT NULL
AND TO_DATE(s.CSIA1_VALUE, 'YYYYMMDD') BETWEEN TRUNC(SYSDATE) AND TRUNC(SYSDATE) + 14;
```

MS SQL Server:

```
SELECT s.CLIENT_ID, s.ITEM_NO, s.CSIA1_VALUE AS PRUEFdatum, s.CURRENT_LOCATION_ID
FROM STOCK_OBJECT s WHERE s.CSIA1_VALUE IS NOT NULL
AND TRY_CAST(s.CSIA1_VALUE AS DATE) BETWEEN CAST(GETDATE() AS DATE) AND DATEADD(day, 14,
CAST(GETDATE() AS DATE));
```

JOB 03: Automatische Werkzeugsperrung bei abgelaufenem Prüfdatum (CSIA 1 Block)

Kategorie: Bestandsführung & Haltbarkeit | **Cron:** 15 0 *** (Täglich 00:15) | **WAMAS-Kontext:** MD002, WCS_STOCKATTRIBUTECHANGE

Sperrt Spezialwerkzeuge und Messmittel automatisch für die Auslagerung (Setzen von stockRating = BLOCKED), sobald das DGUV V3 oder ISO-Prüfdatum in CSIA 1 überschritten ist.

Oracle SQL:

```
INSERT INTO WCS_STOCKATTRIBUTECHANGE (ID, SYS_PARTNER_ID, RECORD_TYPE, CLIENT_ID, ITEM_NO, VARIANT,
STOCK_RATING_FROM, STOCK_RATING_TO, PROCESS_STATE, CREATION_DATE)
SELECT SEQ_WCS_STOCKATTRCHANGE.NEXTVAL, 'CRON_CALIB', 'WCSSTOCKATTRIBUTECHANGE00004', CLIENT_ID,
ITEM_NO, VARIANT, 'AVAILABLE', 'BLOCKED', 'NEW', SYSDATE
FROM STOCK_OBJECT WHERE CSIA1_VALUE IS NOT NULL AND TO_DATE(CSIA1_VALUE, 'YYYYMMDD') < TRUNC(SYSDATE)
AND STOCK_RATING_ID = 'AVAILABLE';
```

MS SQL Server:

```
INSERT INTO WCS_STOCKATTRIBUTECHANGE (SYS_PARTNER_ID, RECORD_TYPE, CLIENT_ID, ITEM_NO, VARIANT,
STOCK_RATING_FROM, STOCK_RATING_TO, PROCESS_STATE, CREATION_DATE)
SELECT 'CRON_CALIB', 'WCSSTOCKATTRIBUTECHANGE00004', CLIENT_ID, ITEM_NO, VARIANT, 'AVAILABLE',
'BLOCKED', 'NEW', GETDATE()
FROM STOCK_OBJECT WHERE CSIA1_VALUE IS NOT NULL AND TRY_CAST(CSIA1_VALUE AS DATE) < CAST(GETDATE() AS
DATE) AND STOCK_RATING_ID = 'AVAILABLE';
```

JOB 04: Meldebestandsprüfung & Nachschubanforderung (Replenishment Trigger)

Kategorie: Bestandsführung & Haltbarkeit | **Cron:** 0 */2 *** (Alle 2 Stunden) | **WAMAS-Kontext:** LM175, LM119, LM080

Überprüft die kumulierten Bestände im SSI LOGIMAT gegen die eingestellten Mindestmengen (Meldebestand). Generiert bei Unterschreitung automatisch Nachschub-Einlageraufträge (Storage Demands) aus dem Pufferlager.

Oracle SQL:

```
SELECT i.CLIENT_ID, i.ITEM_NO, NVL(SUM(s.QTY_AVAILABLE),0) AS ACT_QTY, i.MIN_QTY
FROM ITEM i LEFT JOIN STOCK_OBJECT s ON i.CLIENT_ID = s.CLIENT_ID AND i.ITEM_NO = s.ITEM_NO
GROUP BY i.CLIENT_ID, i.ITEM_NO, i.MIN_QTY HAVING NVL(SUM(s.QTY_AVAILABLE),0) < i.MIN_QTY;
```

MS SQL Server:

```
SELECT i.CLIENT_ID, i.ITEM_NO, ISNULL(SUM(s.QTY_AVAILABLE),0) AS ACT_QTY, i.MIN_QTY
FROM ITEM i LEFT JOIN STOCK_OBJECT s ON i.CLIENT_ID = s.CLIENT_ID AND i.ITEM_NO = s.ITEM_NO
GROUP BY i.CLIENT_ID, i.ITEM_NO, i.MIN_QTY HAVING ISNULL(SUM(s.QTY_AVAILABLE),0) < i.MIN_QTY;
```

JOB 05: Nullbestands- & Fachbereinigung (Compartment Empty Check)

Kategorie: Bestandsführung & Haltbarkeit | **Cron:** 30 1 *** (Täglich 01:30) | **WAMAS-Kontext:** LM002, MD001, STOCK_OBJECT

Findet leere Bestandseinträge (Menge = 0) und entfernt verwaiste Bestands-IDs, um Fächer im Lift wieder als vollständig frei für Neu-Einlagerungen zu kennzeichnen.

Oracle SQL:

```
DELETE FROM STOCK_OBJECT WHERE QTY_AVAILABLE = 0 AND QTY_RESERVED = 0 AND PROCESS_STATE = 'COMPLETED'
AND LAST_MODIFIED < SYSDATE - 7;
```

MS SQL Server:

```
DELETE FROM STOCK_OBJECT WHERE QTY_AVAILABLE = 0 AND QTY_RESERVED = 0 AND PROCESS_STATE = 'COMPLETED'
AND LAST_MODIFIED < DATEADD(day, -7, GETDATE());
```

JOB 06: Bereinigung verarbeiteter Eingangs-Telegramme (inSPM Purge)**Kategorie:** Schnittstellen & Telegramme | **Cron:** 0 2 * * * (Täglich 02:00) | **WAMAS-Kontext:** DI002, SE010, WAMAS2INSPM

Löscht erfolgreich verarbeitete Eingangstelegramme (Status PROCESSED) aus der Tabelle WAMAS2INSPM, die älter als das eingestellte Aufbewahrungsintervall (z. B. 14 Tage) sind.

Oracle SQL:

```
DELETE FROM WAMAS2INSPM WHERE PROCESS_STATE = 'PROCESSED' AND CREATION_DATE < SYSDATE - 14;
```

MS SQL Server:

```
DELETE FROM WAMAS2INSPM WHERE PROCESS_STATE = 'PROCESSED' AND CREATION_DATE < DATEADD(day, -14, GETDATE());
```

JOB 07: Bereinigung verarbeiteter Ausgangs-Telegramme (outSPM Purge)**Kategorie:** Schnittstellen & Telegramme | **Cron:** 15 2 * * * (Täglich 02:15) | **WAMAS-Kontext:** DI003, SE010, WAMAS2OUTSPM

Entfernt quittierte und erfolgreich an das übergeordnete ERP/WMS gesendete Ausgangstelegramme aus den Ausgangspuffern zur Entlastung der Schnittstellentabellen.

Oracle SQL:

```
DELETE FROM WAMAS2OUTSPM WHERE PROCESS_STATE = 'SENT' AND CREATION_DATE < SYSDATE - 14;
```

MS SQL Server:

```
DELETE FROM WAMAS2OUTSPM WHERE PROCESS_STATE = 'SENT' AND CREATION_DATE < DATEADD(day, -14, GETDATE());
```

JOB 08: Schnittstellen-Dateivolumen Archivbereinigung (File Archive Cleanup)**Kategorie:** Schnittstellen & Telegramme | **Cron:** 0 3 * * 0 (Sonntag 03:00) | **WAMAS-Kontext:** LM163, DI014, Dateisystem

Löscht importierte Excel- (.xlsx), CSV- und XML-Dateien aus den lokalen Archivverzeichnissen des WAMAS-Servers (/opt/wamas/interface/archive/), die älter als 30 Tage sind.

Oracle SQL:

```
# Shell Skript Ausführung:
find /opt/wamas/interface/archive/ -type f -mtime +30 -name "*.xlsx" -delete
find /opt/wamas/interface/archive/ -type f -mtime +30 -name "*.xml" -delete
```

MS SQL Server:

```
# PowerShell Skript Ausführung:
Get-ChildItem -Path 'C:\wamas\interface\archive' -Recurse | Where-Object {$_.LastWriteTime -lt (Get-Date).AddDays(-30)} | Remove-Item
```

JOB 09: Schnittstellen-Fehler-Notifizierung & Reset (Stuck Telegram Alert)**Kategorie:** Schnittstellen & Telegramme | **Cron:** */30 * * * * (Alle 30 Min.) | **WAMAS-Kontext:** DI002, DI003, E-Mail-Service

Überprüft, ob Telegramme im Status ERROR vorliegen oder sich seit mehr als 60 Minuten im Status PROCESSING befinden, und versendet einen Alarm an die IT-Administration.

Oracle SQL:

```
SELECT ID, SYS_PARTNER_ID, RECORD_TYPE, ERROR_CODE, ERROR_TEXT
FROM WAMAS2INSPM WHERE PROCESS_STATE = 'ERROR' AND LAST_MODIFIED > SYSDATE - (1/24);
```

MS SQL Server:

```
SELECT ID, SYS_PARTNER_ID, RECORD_TYPE, ERROR_CODE, ERROR_TEXT
FROM WAMAS2INSPM WHERE PROCESS_STATE = 'ERROR' AND LAST_MODIFIED > DATEADD(hour, -1, GETDATE());
```

JOB 10: Historische Einlageraufträge bereinigen (Storage Demand History Purge)**Kategorie:** Auftrags- & Historienverwaltung | **Cron:** 30 2 * * * (Täglich 02:30) | **WAMAS-Kontext:** LM080, LM081, STORAGE_DEMAND

Archiviert und löscht abgeschlossene Einlageraufträge (Storage Demands im Status COMPLETED oder CANCELED) nach Ablauf der Aufbewahrungsfrist.

Oracle SQL:

```
DELETE FROM STORAGE_DEMAND WHERE DEMAND_STATE IN ('COMPLETED', 'CANCELED') AND LAST_MODIFIED <
SYSDATE - 30;
```

MS SQL Server:

```
DELETE FROM STORAGE_DEMAND WHERE DEMAND_STATE IN ('COMPLETED', 'CANCELED') AND LAST_MODIFIED <
DATEADD(day, -30, GETDATE());
```

JOB 11: Historische Kommissionieraufträge bereinigen (Picking Demand History Purge)**Kategorie:** Auftrags- & Historienverwaltung | **Cron:** 45 2 * * * (Täglich 02:45) | **WAMAS-Kontext:** LM071, LM072, PICKING_DEMAND

Bereinigt verarbeitete Kommissionier- und Auslageraufträge (Picking Demands) aus den aktiven Tabellen, um den Hauptspeicher und Index-Größen schlank zu halten.

Oracle SQL:

```
DELETE FROM PICKING_DEMAND WHERE DEMAND_STATE IN ('COMPLETED', 'CANCELED') AND LAST_MODIFIED <
SYSDATE - 30;
```

MS SQL Server:

```
DELETE FROM PICKING_DEMAND WHERE DEMAND_STATE IN ('COMPLETED', 'CANCELED') AND LAST_MODIFIED <
DATEADD(day, -30, GETDATE());
```

JOB 12: Historische Inventuraufträge bereinigen (Inventory Demand History Purge)**Kategorie:** Auftrags- & Historienverwaltung | **Cron:** 0 3 * * * (Täglich 03:00) | **WAMAS-Kontext:** LM090, LM091, INVENTORY_DEMAND

Entfernt quitierte Zähl- und Inventuraufträge aus dem System. Sicherung aller Zählergebnisse in Archiv-Tabellen zur Revisionssicherheit.

Oracle SQL:

```
DELETE FROM INVENTORY_DEMAND WHERE DEMAND_STATE = 'COMPLETED' AND LAST_MODIFIED < SYSDATE - 90;
```

MS SQL Server:

```
DELETE FROM INVENTORY_DEMAND WHERE DEMAND_STATE = 'COMPLETED' AND LAST_MODIFIED < DATEADD(day, -90,
GETDATE());
```

JOB 13: Freigabe verwaister Bediener-Sperren (Stale Session & Lock Release)**Kategorie:** Auftrags- & Historienverwaltung | **Cron:** 0 * * * * (Stündlich) | **WAMAS-Kontext:** LM002, UM002, TERMINAL_LOCK

Hebt temporäre Tablar- und Fachreservierungen am Touch-Client auf, wenn ein Bediener die Sitzung ohne ordnungsgemäße Quittierung abgebrochen hat (Timeout > 30 Min).

Oracle SQL:

```
UPDATE TRAY_ACCESS_OPENING SET LOCK_STATE = 'UNLOCKED', LOCKED_BY_USER = NULL WHERE LOCK_STATE =
'LOCKED' AND LAST_MODIFIED < SYSDATE - (30/1440);
```

MS SQL Server:

```
UPDATE TRAY_ACCESS_OPENING SET LOCK_STATE = 'UNLOCKED', LOCKED_BY_USER = NULL WHERE LOCK_STATE =
'LOCKED' AND LAST_MODIFIED < DATEADD(minute, -30, GETDATE());
```

JOB 14: Put-2-Print Spooler-Bereinigung (Print Spooler Cleanup)**Kategorie:** Druck & Dokumentenmanagement | **Cron:** 30 3 * * * (Täglich 03:30) | **WAMAS-Kontext:** RM011, RM017, Put-2-Print

Löscht temporär generierte ZPL-Etikettendateien und PDF-Quittungsbelege aus dem lokalen Print-Spooler-Verzeichnis des LOGIMAT-Terminals.

Oracle SQL:

```
# Shell Skript:
find /var/spool/wamas/print/ -type f -mtime +2 -delete
```

MS SQL Server:

```
# PowerShell Skript:
Get-ChildItem -Path 'C:\wamas\spool\print' | Where-Object {$_.LastWriteTime -lt
(Get-Date).AddDays(-2)} | Remove-Item
```

JOB 15: Bereinigung temporärer Berichtsdateien (Report Engine Temp Purge)**Kategorie:** Druck & Dokumentenmanagement | **Cron:** 45 3 * * * (Täglich 03:45) | **WAMAS-Kontext:** RM040, BDeploy, Temp Directory

Bereinigt vom WAMAS Reporting System temporär erzeugte PDF-Dateien (z. B. Bestandsberichte `WLSSTOCKREQ002`), um Festplattenplatz zu sparen.

Oracle SQL:

```
# Shell Skript:
find /tmp/wamas_reports/ -type f -mmin +120 -delete
```

MS SQL Server:

```
# PowerShell Skript:
Get-ChildItem -Path 'C:\Windows\Temp\wamas_reports' | Where-Object {$_.LastWriteTime -lt
(Get-Date).AddHours(-2)} | Remove-Item
```

JOB 16: Tägliches Datenbank-Hot-Backup (Database Backup)**Kategorie:** Datenbank & System-Health | **Cron:** 0 2 * * * (Täglich 02:00) | **WAMAS-Kontext:** BDeploy, Oracle Expdp / MSSQL Backup

Führt einen automatischen Online-Dump der WAMAS-Datenbankschemata (wamas5_prod, wamas5_arch) zur Sicherung auf externe Speichermedien durch.

Oracle SQL:

```
# Oracle Data Pump Dump:
expdp wamas/password@WAMASDB schemas=WAMAS5_PROD directory=BACKUP_DIR dumpfile=wamas_full_%U.dmp
logfile=expdp_wamas.log reuse_dumpfiles=Y
```

MS SQL Server:

```
-- MS SQL Server Backup T-SQL:
BACKUP DATABASE [WAMAS5_PROD] TO DISK = 'D:\Backup\WAMAS5_PROD.bak' WITH NOFORMAT, INIT,
COMPRESSION, STATS = 10;
```

JOB 17: Datenbank Index-Reorganisation (Reindex & Fragment Clean)**Kategorie:** Datenbank & System-Health | **Cron:** 0 4 * * 0 (Sonntag 04:00) | **WAMAS-Kontext:** wamas_db_optimize_oracle.sql / mssql

Führt die mitgelieferten WAMAS-Datenbankoptimierungsskripte aus. Baut stark fragmentierte Indizes neu auf und stellt schnelle Antwortzeiten sicher.

Oracle SQL:

```
# Ausführung des WAMAS-Standard-Skripts:
sqlplus wamas/password@WAMASDB @/opt/wamas/scripts/wamas_db_optimize_oracle.sql
```

MS SQL Server:

```
-- MS SQL Server Index Rebuild:
EXEC sp_MSforeachtable @command1="ALTER INDEX ALL ON ? REBUILD WITH (FILLFACTOR = 80);";
```

JOB 18: Aktualisierung der Optimizer-Statistiken (Update Statistics)**Kategorie:** Datenbank & System-Health | **Cron:** 0 1 * * * (Täglich 01:00) | **WAMAS-Kontext:** DBMS_STATS / UPDATE STATISTICS

Aktualisiert die Datenbank-Tabellenstatistiken, damit der Cost-Based Optimizer (CBO) stets optimale Ausführungspläne für WAMAS-Suchabfragen wählt.

Oracle SQL:

```
BEGIN
DBMS_STATS.GATHER_SCHEMA_STATS('WAMAS5_PROD', cascade => TRUE, options => 'GATHER AUTO');
END;
/
```

MS SQL Server:

```
-- MS SQL Server Update Statistics:
EXEC sp_updatestats;
```

JOB 19: Applikations- & BDeploy-Logfile Rotation (Log Maintenance)**Kategorie:** Datenbank & System-Health | **Cron:** 0 3 * * * (Täglich 03:00) | **WAMAS-Kontext:** logrotate, BDeploy Launcher Logs

Komprimiert Applikations-Logfiles (WildFly, RAP-Server, BDeploy) und löscht alte gepackte Log-Archive (*.gz) nach 30 Tagen.

Oracle SQL:

```
# logrotate Konfiguration (/etc/logrotate.d/wamas):
/var/log/wamas/*.log {
daily
rotate 30
compress
missingok
}
```

MS SQL Server:

```
# PowerShell Log Clean:
Get-ChildItem -Path 'C:\wamas\logs' -Filter '*.log' | Where-Object {$_.LastWriteTime -lt
(Get-Date).AddDays(-30)} | Remove-Item
```

JOB 20: System Health & Connection Watchdog (Interface & PLC Check)**Kategorie:** Datenbank & System-Health | **Cron:** */5 * * * * (Alle 5 Minuten) | **WAMAS-Kontext:** BDeploy, DI002, Network Monitoring

Überprüft kontinuierlich die Netzwerkkonnektivität zu den LOGIMAT-SPSen, Etikettendruckern, Waagen und REST-Endpunkten. Löst bei Verbindungsabbruch Alarm aus.

Oracle SQL:

```
# Python / Shell Connectivity Watchdog:
python3 /opt/wamas/scripts/wamas_health_check.py --ping-plc --check-rest-endpoints --alert-email
```

MS SQL Server:

```
# PowerShell Connectivity Check:
Test-NetConnection -ComputerName '192.168.1.50' -Port 8080
```

Zusammenfassendes Crontab-Beispiel (/etc/crontab)

```
# =====
# WAMAS® Lift & Store 2.8 - System Crontab Übersicht (/etc/crontab)
# =====
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=wamas-admin@company.com

# --- Bestandsführung & Haltbarkeit ---
5 0 * * * wamas /opt/wamas/scripts/run_bbd_block.sh > /dev/null 2>&1
15 0 * * * wamas /opt/wamas/scripts/run_calib_block.sh > /dev/null 2>&1
0 6 * * * wamas /opt/wamas/scripts/run_calib_warning.sh > /dev/null 2>&1
0 */2 * * * wamas /opt/wamas/scripts/run_replenishment_check.sh > /dev/null 2>&1
30 1 * * * wamas /opt/wamas/scripts/run_empty_compartment_clean.sh > /dev/null 2>&1

# --- Schnittstellen & Housekeeping ---
0 2 * * * wamas /opt/wamas/scripts/run_inspm_purge.sh > /dev/null 2>&1
15 2 * * * wamas /opt/wamas/scripts/run_outspm_purge.sh > /dev/null 2>&1
*/30 * * * * wamas /opt/wamas/scripts/run_stuck_telegram_alert.sh > /dev/null 2>&1
0 3 * * * 0 wamas /opt/wamas/scripts/clean_file_archive.sh > /dev/null 2>&1

# --- Auftrags- & Historienverwaltung ---
30 2 * * * wamas /opt/wamas/scripts/purge_storage_demands.sh > /dev/null 2>&1
45 2 * * * wamas /opt/wamas/scripts/purge_picking_demands.sh > /dev/null 2>&1
0 3 * * * wamas /opt/wamas/scripts/purge_inventory_demands.sh > /dev/null 2>&1
0 * * * * wamas /opt/wamas/scripts/release_stale_locks.sh > /dev/null 2>&1

# --- Druck & Dokumentenmanagement ---
30 3 * * * wamas /opt/wamas/scripts/clean_print_spooler.sh > /dev/null 2>&1
45 3 * * * wamas /opt/wamas/scripts/clean_report_temp.sh > /dev/null 2>&1

# --- Datenbank & System-Health ---
0 1 * * * oracle /opt/wamas/scripts/gather_db_stats.sh > /dev/null 2>&1
0 2 * * * oracle /opt/wamas/scripts/oracle_exdpdp_backup.sh > /dev/null 2>&1
0 3 * * * root /usr/sbin/logrotate /etc/logrotate.d/wamas > /dev/null 2>&1
0 4 * * * 0 oracle /opt/wamas/scripts/wamas_db_optimize.sh > /dev/null 2>&1
*/5 * * * * wamas /opt/wamas/scripts/wamas_health_check.py > /dev/null 2>&1
```