The Ultimate Odoo Backup Verification Checklist

Your Complete Guide to Bulletproof Backup Verification

Created by Aria Shaw - The Definitive Odoo Backup Guide

About This Checklist

This comprehensive checklist ensures your Odoo backup strategy actually works when you need it most. A backup you haven't verified is just wishful thinking. This checklist transforms hope into confidence by providing systematic verification procedures.

How to Use This Checklist:

- Print and keep a physical copy near your workstation
- Check off each item as you complete it
- Customize sections for your specific environment
- Use monthly to track improvement over time

Pre-Backup Verification (Do This Before Every Backup)

System Health Check

🔍 Step 1: Verify System Resources

□ **Check available disk space** (at least 2x database size free)

```
df -h /backup/odoo/
```

□ **Confirm PostgreSQL is responding**

```
sudo -u postgres psql -c "SELECT version();"
```

□ **Verify Odoo process is stable** (no recent crashes)

```
systemctl status odoo
```

□ **Check for ongoing maintenance** that might affect backup

```
ps aux | grep -E "(vacuum|reindex|upgrade)"
```

🔍 Step 2: Database Integrity Check

□ Run database consistency check

```
sudo -u postgres psql -d your_database -c "SELECT * FROM pg_stat_activity WHERE state = 'active';"
```

□ Verify no long-running transactions

```
sudo -u postgres psql -d your_database -c "SELECT pid, now() - pg_stat_activity.query_start AS duration, query FROM pg_stat_activity WHERE (now() - pg_stat_activity.query_start) > interval '5 minutes';"
```

□ Check for database bloat warnings

```
sudo -u postgres psql -d your_database -c "SELECT schemaname, tablename, pg_size_pretty(pg_total_relation_size(schemaname||'.'||tablename)) as size FROM pg_tables ORDER BY pg_total_relation_size(schemaname||'.'||tablename) DESC LIMIT 10;"
```

Filestore Verification

🔍 Step 3: Filestore Health Check

□ Verify filestore directory exists and is accessible

```
ls -la /var/lib/odoo/filestore/your_database/
```

□ Check for filestore permissions issues

```
sudo -u odoo ls /var/lib/odoo/filestore/your_database/ >/dev/null
```

□ Confirm no broken symlinks

```
find /var/lib/odoo/filestore/your_database/ -type l -exec test ! -e {} \; -print
```

□ Verify filestore size is reasonable

```
du -sh /var/lib/odoo/filestore/your_database/
```

□ Check for recent filestore errors in logs

```
grep -i "filestore\|attachment" /var/log/odoo/odoo.log | tail -20
```

During Backup Verification (Monitor These While Backup Runs)

Real-Time Monitoring

🔍 Step 4: Active Backup Monitoring

□ Monitor backup process CPU/memory usage

```
top -p $(pgrep -f backup)
```

□ Watch for error messages in real-time

```
tail -f /var/log/odoo/odoo.log | grep -E "(error|warning|backup)"
```

□ Verify backup file is growing at reasonable rate

```
watch "ls -lh /backup/odoo/current_backup*"
```

□ Check PostgreSQL connection count during backup

```
sudo -u postgres psql -c "SELECT count(*) FROM pg_stat_activity;"
```

□ Monitor system load average

```
uptime
```

Early Warning Detection

🔍 Step 5: Detect Problems Early

□ Check for "out of space" warnings

```
df -h | grep -E "(9[0-9]%|100%)"
```

□ **Monitor for memory pressure**

```
free -h | grep -E "Mem:|Swap:"
```

□ **Watch for backup timeout warnings**

```
grep -i timeout /var/log/odoo/odoo.log | tail -5
```

□ **Verify no database locks blocking backup**

```
sudo -u postgres psql -d your_database -c "SELECT pid, mode, locktype,
relation::regclass, transactionid FROM pg_locks WHERE NOT granted;"
```

Post-Backup Verification (Critical - Never Skip This)

File Integrity Checks

🔍 Step 6: Backup File Validation

□ **Verify backup file was created**

```
ls -la /backup/odoo/$(date +%Y%m%d)*
```

□ **Check backup file size is reasonable** (>10MB for real data)

```
BACKUP_SIZE=$(stat -c%s "/backup/odoo/latest_backup.zip")
echo "Backup size: $(($BACKUP_SIZE / 1024 / 1024)) MB"
```

□ **Test ZIP file integrity**

```
unzip -t /backup/odoo/latest_backup.zip
```

□ **Verify manifest.json exists and is valid**

```
unzip -p /backup/odoo/latest_backup.zip manifest.json | python3 -c "import json,
sys; json.load(sys.stdin)"
```

□ Check dump file is not corrupted

```
if [[ -f backup.dump ]]; then pg_restore --list backup.dump >/dev/null; fi
```

Content Verification

🔍 Step 7: Backup Content Validation

□ Verify database dump contains expected tables

```
pg_restore --list backup.dump | grep -c "TABLE DATA"
```

□ Check filestore directory is included

```
unzip -l backup.zip | grep -c "filestore/"
```

□ Verify module data is present

```
unzip -p backup.zip manifest.json | grep -o '"modules": \[[^]]*\]' | wc -c
```

□ Check for custom addon data

```
unzip -l backup.zip | grep -E "custom|local"
```

□ Verify attachment count matches expectations

```
ATTACHMENT_COUNT=$(unzip -p backup.zip dump.sql | grep -c "ir_attachment")
echo "Found $ATTACHMENT_COUNT attachment references"
```

Test Restore Verification (Monthly Recommended)

Staging Environment Test

🔍 Step 8: Complete Restore Test (Monthly)

□ Create isolated test environment

□ Restore backup to test database

```
sudo -u postgres createdb test_restore_$(date +%Y%m%d)
```

□ **Verify restore completes without errors**

```
./restore_backup.sh test_restore_$(date +%Y%m%d) /backup/odoo/latest_backup.zip
```

□ **Test database connectivity**

```
sudo -u postgres psql -d test_restore_$(date +%Y%m%d) -c "SELECT COUNT(*) FROM res_users;"
```

□ **Verify key business data is present**

```
# Replace with your critical business queries
sudo -u postgres psql -d test_restore_$(date +%Y%m%d) -c "SELECT COUNT(*) FROM sale_order WHERE state='done';"
```

Application Function Test

🔍 Step 9: Application-Level Verification

□ **Start Odoo with restored database**

```
sudo -u odoo /opt/odoo/odoo-bin -d test_restore_$(date +%Y%m%d) --test-enable --stop-after-init
```

□ **Test login functionality**

```
curl -X POST http://localhost:8069/web/session/authenticate -d '{"jsonrpc":"2.0","method":"call","params":{"db":"test_restore","login":"admin","password":"admin"}}'
```

□ **Verify file attachments work** *Test by downloading a known attachment*

□ **Check key modules load correctly**

```
grep -E "(installed|upgraded)" /var/log/odoo/odoo.log | tail -10
```

□ **Test basic workflow** (create/edit record)

□ **Clean up test environment**

```
sudo -u postgres dropdb test_restore_$(date +%Y%m%d)
```

Automation Verification (For Automated Backups)

Script Health Check

🔍 Step 10: Backup Automation Verification

□ **Verify cron job is running**

```
crontab -l | grep backup
```

□ **Check last backup completion status**

```
grep -i "backup.*complete" /var/log/odoo/backup.log | tail -1
```

□ **Verify notification system works**

```
echo "Test backup notification" | mail -s "Backup Test" admin@yourcompany.com
```

□ **Check backup retention is working**

```
find /backup/odoo/ -mtime +30 | wc -l  # Should be managed by retention policy
```

□ **Verify cloud upload** (if configured)

```
aws s3 ls s3://your-backup-bucket/$(date +%Y/%m/%d)/ | wc -l
```

Monitoring and Alerting

🔍 Step 11: Alert System Verification

□ **Test backup failure notifications** *Temporarily break backup script and verify alerts work*

□ **Verify monitoring dashboard updates**

```
curl -s http://your-server/backup-status.html | grep "$(date +%Y-%m-%d)"
```

□ Check log rotation is working

```
ls -la /var/log/odoo/backup.log*
```

□ Test recovery contact list is current

□ Verify escalation procedures work

Emergency Preparedness Check (Quarterly)

Disaster Recovery Readiness

🔍 Step 12: Disaster Recovery Preparedness (Quarterly)

□ Document current backup locations

□ Verify backup accessibility from different locations

□ Test backup download from cloud storage

```
aws s3 cp s3://your-backup-bucket/latest-backup.zip /tmp/test-download.zip
```

□ Confirm team knows recovery procedures

□ Verify contact information is current

□ Test communication channels during simulated outage

□ Review and update recovery time objectives (RTO)

□ Check recovery point objectives (RPO) are being met

Backup Quality Scoring

Use the Backup Quality Score Calculator script to evaluate your backup strategy.

Scoring Categories:

- **Pre-backup verification:** 20 points
- **During backup monitoring:** 15 points
- **Post-backup verification:** 25 points
- **Test restore procedures:** 30 points
- **Automation and monitoring:** 20 points
- **Emergency preparedness:** 10 points

Total Possible Score: 120 points

Score Interpretation:

- **100-120: 🏆 EXCELLENT** - Enterprise-grade backup strategy
- **80-99: ✅ GOOD** - Solid practices with room for improvement
- **60-79: ⚠️ ADEQUATE** - Basic coverage, needs strengthening
- **40-59: ❌ POOR** - Significant gaps in backup strategy
- **0-39: 🚨 CRITICAL** - Backup strategy needs immediate attention

---

Pro Tips for Using This Checklist

1. **Print and Laminate**: Keep a physical copy near your workstation for emergencies

2. **Customize for Your Environment**: Add specific checks for your custom modules and business requirements

3. **Track Trends**: Keep a log of scores over time to identify improvement patterns

4. **Share with Team**: Ensure everyone knows the verification procedures

5. **Automate What You Can**: Turn manual checks into scripts where possible

6. **Regular Review**: Update this checklist as your system evolves

7. **Document Exceptions**: Note any checks that don't apply to your setup

8. **Practice Under Pressure**: Run through the checklist during simulated emergencies

---

Notes Section

Use this space to record environment-specific information:

Database Names:

- Production: _______________
- Staging: _______________
- Development: _______________

Backup Locations:

- Local: _______________
- Cloud: _______________
- Offsite: _______________

Key Contacts:

- Primary Admin: _______________
- Backup Admin: _______________
- Emergency Contact: _______________

Custom Checks for Your Environment: □ ______________________________ □ ______________________________ □ ______________________________ □ ______________________________ □ ______________________________

Checklist History

Track your backup quality improvements:

Date	Score	Notes
	/120	
	/120	
	/120	
	/120	
	/120	

Remember: A backup you haven't verified is just wishful thinking. This checklist transforms hope into confidence by ensuring your backup strategy actually works when you need it most.

This checklist is part of "The Definitive Guide to Odoo Database Backup and Restore" by Aria Shaw. For updates and additional resources, visit: https://ariashaw.github.io