✅ Odoo Deployment Checklist

Use this checklist every time you deploy code to production

📋 Pre-Deployment Checklist

Code Review

- [ ] All code changes reviewed by at least one team member
- [ ] Pull Request approved and merged to `development` branch
- [ ] No merge conflicts
- [ ] Code follows team coding standards
- [ ] Comments and documentation updated

Testing

- [ ] Feature tested locally on development server
- [ ] No errors in Odoo logs during testing
- [ ] All affected modules load correctly
- [ ] Database migrations tested (if any)
- [ ] No broken dependencies

Backup

- [ ] Database backup created

```
sudo -u postgres pg_dump odoo18 > /tmp/odoo18_backup_$(date +%Y%m%d_%H%M%S).sql
```

- [ ] Backup verified (check file size > 0)
- [ ] Note backup location: `/tmp/odoo18_backup_YYYYMMDD_HHMMSS.sql`

Communication

- [ ] Team notified of upcoming deployment
- [ ] Deployment time scheduled (avoid peak hours)
- [ ] Rollback plan discussed if needed

🚀 Deployment Steps

Step 1: Connect to Server

```
ssh -i "MyTestApp-KeyPair.pem" ec2-user@56.228.2.47
```

- [ ] Successfully connected to server

Step 2: Navigate to Project

```
cd /home/ec2-user/odoo18
```

- [ ] In correct directory (`pwd` shows `/home/ec2-user/odoo18`)

Step 3: Check Current Status

```
git status
git branch
```

- [ ] On `development` branch
- [ ] Working directory clean (no uncommitted changes)

Step 4: Pull Latest Code

```
git pull origin development
```

- [ ] Pull successful (no errors)
- [ ] Note commit hash: `git log --oneline -1`
- [ ] Commit hash: _____

Step 5: Review Changes

```
git log --oneline -5
git diff HEAD~1 --name-only
```

- [ ] Reviewed what files changed
- [ ] No unexpected changes

Step 6: Check for Database Updates

```
grep -r "def _auto_init" --include="*.py" | head -5
```

- [ ] Database schema changes identified (if any)
- [ ] Migration plan ready (if needed)

Step 7: Restart Odoo

```
sudo systemctl restart odoo
```

- [ ] Restart command executed successfully

Step 8: Monitor Startup

```
sudo systemctl status odoo
```

- [ ] Service shows "active (running)"
- [ ] No immediate errors

Step 9: Check Logs

```
tail -n 100 /home/ec2-user/.odoo/odoo-server.log
```

- [ ] No ERROR or CRITICAL messages
- [ ] Odoo started successfully
- [ ] All modules loaded
- [ ] HTTP service running on port 8069

Step 10: Verify Web Access

Open browser: `http://56.228.2.47:8069`

- [ ] Odoo web interface loads
- [ ] Login works
- [ ] Main dashboard displays correctly

🧪 Post-Deployment Testing

Smoke Tests

- [ ] Navigate to home page
- [ ] Test login/logout
- [ ] Check recently modified modules
- [ ] Test key workflows:
  - [ ] Create new record
  - [ ] Edit existing record
  - [ ] Search functionality
  - [ ] Reports generation

Module-Specific Tests

For each changed module:

- [ ] Module: _____
  - [ ] Feature works as expected
  - [ ] No console errors
  - [ ] Data displays correctly

Performance Check

```
# Check server load
top
# Press 'q' to exit

# Check memory
free -h
```

```
# Check disk space
df -h
```

- [ ] CPU usage normal (< 80%)
- [ ] Memory usage acceptable
- [ ] Disk space sufficient (> 20% free)

📊 Monitoring (First 30 Minutes)

Continuous Log Monitoring

```
tail -f /home/ec2-user/.odoo/odoo-server.log
```

Watch for:

- [ ] No repeated errors
- [ ] No database errors
- [ ] No permission errors
- [ ] Normal request processing

Check Service Status

```
# Every 10 minutes
sudo systemctl status odoo
```

- [ ] 10 min: Service still running
- [ ] 20 min: Service still running
- [ ] 30 min: Service still running

User Monitoring

- [ ] Check with team if they can access Odoo
- [ ] No user-reported issues
- [ ] Normal operation confirmed

❌ Rollback Procedure (If Issues Found)

Quick Rollback

```
# 1. Note current commit
git log --oneline -1

# 2. Find last working commit
git log --oneline -10

# 3. Rollback to previous commit
git reset --hard PREVIOUS_COMMIT_HASH
```

```
# 4. Restart Odoo
sudo systemctl restart odoo

# 5. Verify it works
tail -f /home/ec2-user/.odoo/odoo-server.log
```

Database Rollback (If Needed)

```
# 1. Stop Odoo
sudo systemctl stop odoo

# 2. Restore database
sudo -u postgres psql odoo18 < /tmp/odoo18_backup_YYYYMMDD_HHMMSS.sql

# 3. Start Odoo
sudo systemctl start odoo

# 4. Monitor logs
tail -f /home/ec2-user/.odoo/odoo-server.log
```

Notify Team

- [ ] Team notified of rollback
- [ ] Issue documented in GitHub
- [ ] Root cause analysis scheduled

📝 Post-Deployment Documentation

Update Deployment Log

Record in `DEPLOYMENT_LOG.md`:

```
## Deployment YYYY-MM-DD HH:MM

- **Deployed by**: Your Name
- **Commit**: abc1234
- **Features**: Brief description
- **Issues**: None / List issues
- **Status**: ✅ Success / ❌ Rolled back
```

GitHub

- [ ] Close related issues
- [ ] Update project board
- [ ] Tag release (if major deployment)

```
git tag v1.0.0
```

```
git push origin v1.0.0
```

Team Communication

- [ ] Notify team of successful deployment
- [ ] Share release notes (if applicable)
- [ ] Document any known issues or workarounds

🔍 Common Issues & Solutions

Issue: Odoo won't start

Check:

```
sudo journalctl -u odoo -n 50
```

Common causes:

- Python syntax error → Fix code and redeploy
- Missing dependency → Install with `sudo pip3.11 install package`
- Database connection → Check PostgreSQL is running

Issue: Module not loading

Check:

```
grep "module_name" /home/ec2-user/.odoo/odoo-server.log
```

Solution:

```
# Update module via Odoo UI
# Apps → Search module → Upgrade
# Or via command line:
./odoo-bin -u module_name -d odoo18
```

Issue: Permission errors

Check:

```
ls -la /home/ec2-user/odoo18/addons/custom_module
```

Solution:

```
sudo chown -R ec2-user:ec2-user /home/ec2-user/odoo18
```

Issue: Port already in use

Check:

```
sudo netstat -tlnp | grep 8069
```

Solution:

```
# Kill the process using the port
sudo kill -9 PROCESS_ID
sudo systemctl restart odoo
```

📞 Emergency Contacts

Situation	Contact
Deployment fails	@karem505
Database issues	DBA Team
Server down	DevOps Team
Critical bug	Development Lead

📊 Deployment Metrics

Track These Metrics

- **Deployment duration**: ____ minutes
- **Downtime**: ____ minutes (if any)
- **Issues found**: ____ (number)
- **Rollbacks**: ____ (number)

Goal Metrics

- ✅ Deployment < 5 minutes
- ✅ Downtime < 30 seconds
- ✅ Zero issues
- ✅ Zero rollbacks

✅ Final Sign-Off

Deployment Date: _____ **Deployed By**: _____ **Commit Hash**: _____ **Status**: ☐ Success ☐ Issues ☐ Rolled Back

Sign-off:

- [ ] All tests passed
- [ ] No errors in logs
- [ ] Users can access system
- [ ] Monitoring shows normal operation
- [ ] Documentation updated
- [ ] Team notified

Notes:

Reminder: Keep this checklist handy for every deployment!

Print this: Or keep it open in a browser tab during deployment.