CUBE PRO

On-Premise Deployment & Release Management Guide

Version: 1.0
Date: August 27, 2025
Document Type: Implementation Guide
Target Audience: System Administrators, DevOps Engineers

Table of Contents

1. Implementation Overview

This comprehensive guide provides step-by-step instructions for implementing the CUBE PRO work order management system on an on-premise server infrastructure.

Key Benefits

- Enterprise-grade reliability for mission-critical operations
- Scalable architecture supporting growth from small teams to enterprises
- Security-first approach with comprehensive protection measures
- Zero data loss through robust backup and recovery procedures
- Minimal downtime using blue-green deployment strategies

Scope of Implementation

- Complete server setup and configuration procedures
- Multi-environment deployment (Development, Staging, Production)
- Automated backup and recovery systems
- Security hardening and best practices
- Monitoring and maintenance protocols

2. System Requirements

Minimum Hardware Requirements

- CPU: 4 cores (Intel Xeon or AMD EPYC recommended)
- RAM: 8GB minimum, 16GB recommended for production
- Storage: 500GB SSD for OS, application, and database
- Network: Gigabit Ethernet connection
- Backup Storage: Additional 1TB for backup retention

Recommended Production Hardware

- CPU: 8+ cores for optimal performance
- RAM: 32GB or higher for heavy workloads
- Storage: 1TB+ NVMe SSD for fast I/O operations
- Network: Redundant Gigabit connections for high availability
- Backup: Network-attached storage (NAS) or dedicated backup server

Software Requirements

- Operating System: Ubuntu Server 22.04 LTS (Primary) or CentOS/RHEL 8/9

- Database: PostgreSQL 14+ for data persistence
- Web Server: Nginx for reverse proxy and static file serving
- Application Server: Gunicorn for Python WSGI applications
- Cache: Redis for session management and caching
- Version Control: Git for source code management

3. Installation Steps

Step 1: Operating System Preparation

Update the system and install essential packages for the CUBE environment.

Step 2: User Account Setup

Create a dedicated application user with appropriate permissions for security isolation.

Step 3: Directory Structure Creation

Establish organized directory structure for different environments and operational needs.

Step 4: Database Server Installation

Install and configure PostgreSQL database server with security hardening.

Step 5: Web Server Configuration

Set up Nginx as reverse proxy with SSL/TLS termination and static file serving.

Step 6: Application Dependencies

Install Python runtime, virtual environments, and required application dependencies.

4. Environment Setup

Development Environment

Isolated environment for feature development and initial testing with relaxed security for debugging.

Staging Environment

Production-like environment for integration testing with production data copies and realistic load testing.

Production Environment

Live environment serving end users with maximum security, monitoring, and performance optimization.

Environment Isolation

- Separate databases for each environment
- Independent configuration files and secrets
- Isolated network access and firewall rules
- Environment-specific logging and monitoring
- Separate backup and recovery procedures

5. Database Configuration

PostgreSQL Installation and Setup

Install PostgreSQL database server with optimized configuration for CUBE workloads.

Database Security Configuration

- Encrypted connections using SSL/TLS
- Role-based access control with principle of least privilege
- Regular security updates and vulnerability patching
- Audit logging for compliance and security monitoring
- Backup encryption for data protection at rest

Performance Optimization

- Memory allocation tuning for optimal query performance
- Index optimization for frequently accessed data
- Connection pooling to manage database connections efficiently
- Query performance monitoring and optimization
- Regular database maintenance and statistics updates

6. Deployment Strategy

Blue-Green Deployment Model

Zero-downtime deployment strategy ensuring business continuity during updates.

Deployment Process

- Blue Environment: Current production serving live traffic
- Green Environment: New version preparation and testing
- Health Validation: Comprehensive testing before traffic switch
- Traffic Switch: Instantaneous routing from blue to green
- Monitoring: Real-time validation of successful deployment
- Rollback: Immediate reversion capability if issues detected

Release Management

- Feature development in dedicated branches
- Integration testing in staging environment
- Automated testing and quality assurance
- Staged rollout with monitoring and validation
- Documentation and change management

7. Backup and Recovery

Automated Backup System

Comprehensive backup strategy ensuring data protection and business continuity.

Backup Schedule

- Daily: Database backups at 2 AM with 30-day retention
- Weekly: Full application backups every Sunday with 12-week retention
- Monthly: Complete system snapshots with long-term archival

Recovery Procedures

- Recovery Time Objective (RTO): 4 hours maximum downtime
- Recovery Point Objective (RPO): 24 hours maximum data loss
- Automated backup verification and integrity testing
- Documented recovery procedures with regular testing
- Disaster recovery site preparation and maintenance

8. Security Implementation

Server Security Hardening

- Firewall configuration with minimal open ports
- SSL/TLS encryption for all communications
- SSH key-based authentication with disabled password login
- Regular security updates and vulnerability management
- Intrusion detection and prevention systems

Application Security

- Environment variable encryption for sensitive data
- Database connection security with encrypted credentials
- Session management with secure token handling
- Input validation and sanitization against injection attacks
- Cross-Site Request Forgery (CSRF) protection

Access Control

- Role-based access control (RBAC) implementation
- Strong password policies and complexity requirements
- Multi-factor authentication (MFA) for administrative access
- Comprehensive audit logging and monitoring
- Regular access reviews and permission audits