CUBE PRO

On-Premise Deployment Guide

Version 2.0 - Comprehensive Implementation Guide
Generated: August 27, 2025
Target: System Administrators and DevOps Engineers

Complete Enterprise Deployment Solution featuring:

- Zero-downtime blue-green deployment strategy
- Comprehensive backup and disaster recovery procedures
- Multi-environment setup (Development, Staging, Production)
- Advanced security hardening and monitoring
- Automated maintenance and health monitoring
- Scalable architecture for enterprise growth

Table of Contents

1. System Requirements and Prerequisites

Hardware Requirements

- CPU: Minimum 4 cores, recommended 8+ cores for production workloads
- RAM: Minimum 8GB, recommended 16GB for development, 32GB+ for production
- Storage: Minimum 500GB SSD, recommended 1TB+ NVMe SSD for optimal performance
- Network: Gigabit Ethernet, redundant connections recommended for production
- Backup Storage: Additional 1TB minimum for backup retention and disaster recovery

Software Requirements

- Operating System: Ubuntu Server 22.04 LTS (primary) or CentOS/RHEL 8/9
- Database: PostgreSQL 14 or later with SSL support
- Web Server: Nginx with SSL/TLS capabilities
- Application Runtime: Python 3.9+ with virtual environment support
- Cache Server: Redis 6+ for session management and application caching
- Version Control: Git 2.30+ for source code management
- Process Manager: Systemd for service management
- Backup Tools: PostgreSQL utilities, rsync, and compression tools

2. Server Setup and Installation Process

Step 1: Operating System Preparation

Update the system and install essential packages:

```
# Update system packages
sudo apt update && sudo apt upgrade -y

# Install essential packages
sudo apt install -y python3 python3-pip python3-venv nginx postgresql \
postgresql-contrib redis-server git curl wget htop unzip

# Install security tools
sudo apt install -y ufw fail2ban
```

Step 2: User Account and Directory Setup

```
# Create dedicated application user
sudo useradd -m -s /bin/bash cubeapp
sudo usermod -aG sudo cubeapp

# Create directory structure
sudo mkdir -p /home/cubeapp/{cube-prod,cube-dev,cube-staging,backups,logs,scripts}
sudo chown -R cubeapp:cubeapp /home/cubeapp/
```

3. Database Installation and Configuration

PostgreSQL Setup

Configure PostgreSQL for optimal performance and security:

```
# Start and enable PostgreSQL
sudo systemctl start postgresql
sudo systemctl enable postgresql

# Create databases and user
sudo -u postgres psql <<EOF
CREATE DATABASE cube_prod;
CREATE DATABASE cube_dev;
CREATE DATABASE cube_staging;
CREATE USER cubeapp WITH PASSWORD  secure_password_here ;
GRANT ALL PRIVILEGES ON DATABASE cube_prod TO cubeapp;
GRANT ALL PRIVILEGES ON DATABASE cube_dev TO cubeapp;
GRANT ALL PRIVILEGES ON DATABASE cube_staging TO cubeapp;
EOF
```

4. Security Implementation

- Firewall Configuration: Configure UFW to allow only necessary ports (22, 80, 443)
- SSL/TLS Setup: Install Let s Encrypt certificates for secure communications
- Database Security: Enable SSL connections and configure authentication
- Application Security: Use environment variables for secrets and credentials
- System Hardening: Disable unused services and apply security updates
- Access Control: Implement role-based permissions and strong passwords
- Monitoring: Set up intrusion detection and log monitoring systems

5. Blue-Green Deployment Strategy

Overview

The blue-green deployment strategy ensures zero-downtime updates by maintaining two identical production environments. During deployment, traffic is switched from the current environment (blue) to the updated environment (green) after successful testing and validation.

Deployment Process

- Prepare green environment with new application version
- Run comprehensive tests including database migrations
- Perform health checks and validation procedures
- Switch load balancer traffic from blue to green environment
- Monitor application performance and error rates

- Maintain blue environment for immediate rollback if needed
- Decommission blue environment after successful validation period

6. Backup and Disaster Recovery

Backup Strategy

- Daily automated database backups at 2 AM with 30-day retention
- Weekly full application backups every Sunday with 12-week retention
- Monthly system snapshots for long-term disaster recovery
- Real-time database replication for high-availability configurations
- Automated backup verification and integrity testing procedures
- Offsite backup storage for geographic disaster protection

Recovery Objectives

Recovery Time Objective (RTO): Maximum 4 hours for full system restoration Recovery Point Objective (RPO): Maximum 24 hours of data loss in worst-case scenarios High Availability Target: 99.9% uptime for production systems