

Infrastructure Deployment Checklist 2026

Tobias Weiss | tobias-weiss.org | [Graphwiz.AI](https://graphwiz.ai)

2026-07-25

? Infrastructure Deployment Checklist 2026

20 Points for Production-Grade Open-Source Infrastructure

By Tobias Weiss / tobias-weiss.org | @tobiasweiss_ai

? About This Checklist

This checklist is designed to help you deploy a **secure, reliable, and scalable** open-source infrastructure stack. It covers the essential components from our [Open-Source Infrastructure Stack Guide](#):

- **Reverse Proxy** (Nginx/Traefik + Certbot + CrowdSec)
- **Data Layer** (PostgreSQL, Redis, MinIO)
- **Workflow Automation** (n8n)
- **Observability** (Prometheus, Grafana, Loki, Alloy)

Use this checklist to ensure you haven't missed any critical steps when deploying or managing your production environment.

? General Deployment

- ☐ **Dedicated Server:** Deploy on a dedicated Linux server (VM or bare metal) with a public IP
 - ☐ **Docker Engine:** Install Docker Engine 24+ with Docker Compose v2
 - ☐ **Firewall:** Configure host firewall (UFW/firewalld) to only allow essential inbound ports (22, 80, 443)
 - ☐ **SSH Hardening:** Disable root SSH login, use key-based authentication, disable password authentication
 - ☐ **System Updates:** Keep OS and Docker updated (e.g., automated `apt update && apt upgrade`)
 - ☐ **Time Sync:** Ensure NTP is configured for accurate time (critical for logs/metrics)
 - ☐ **File System:** Use modern filesystems (ext4/XFS) and consider LVM for flexibility
 - ☐ **Backup Strategy:** Implement a comprehensive backup strategy for all persistent data
 - ☐ **Monitoring:** Set up host-level monitoring (CPU, memory, disk I/O, network)
 - ☐ **Resource Limits:** Define CPU/memory limits for all containers in `docker-compose.yml`
-

? Reverse Proxy (Nginx / Traefik)

- ☐ **SSL/TLS:** Enable HTTPS with Let's Encrypt certificates (automated renewal)
- ☐ **HSTS:** Implement HTTP Strict Transport Security (HSTS) header
- ☐ **WAF Integration:** Integrate CrowdSec or similar Web Application Firewall (WAF) for bot/attack mitigation
- ☐ **Rate Limiting:** Configure rate limiting for common endpoints (login, API, webhooks)
- ☐ **Secure Headers:** Implement security headers (CSP, X-Frame-Options, X-Content-Type-Options)

- ☐ **Logging:** Configure access and error logging, ensure logs are forwarded to observability
- ☐ **Health Checks:** Configure upstream health checks for backend services
- ☐ **Caching:** Implement proxy caching for static assets (if applicable)
- ☐ **DDoS Protection:** Consider Cloudflare/similar for additional DDoS protection (if public-facing)
- ☐ **HTTP/2:** Enable HTTP/2 for performance

? **Related Stack:** [SSL Reverse Proxy Stack ->](#)

? Data Layer (PostgreSQL, Redis, MinIO)

PostgreSQL

- ☐ **Configuration:** Tune `postgresql.conf` for production (`shared_buffers`, `work_mem`, `effective_cache_size`)
- ☐ **Authentication:** Use strong passwords and/or client certificates
- ☐ **Backups:** Automated daily `pg_dump` with compression, 30-day retention
- ☐ **Off-site Backup:** Configure S3 sync (e.g., to MinIO) for off-site disaster recovery
- ☐ **WAL Archiving:** Enable WAL archiving for point-in-time recovery (for critical apps)
- ☐ **Connection Pooling:** Use a connection pooler (PgBouncer) for high concurrency
- ☐ **Replication:** Set up streaming replication for high availability (primary/standby)
- ☐ **Monitoring:** Collect PostgreSQL metrics (e.g., `pg_exporter` for Prometheus)
- ☐ **Disk IO:** Use SSD/NVMe for optimal performance

? **Related Stack:** [Database Foundation Stack ->](#)

Redis

- ☐ **Persistence:** Enable both RDB snapshots and AOF append-only log for durability
- ☐ **Security:** Password-protect Redis instance, bind to specific IP/interface
- ☐ **Resource Limits:** Set `maxmemory` and `maxmemory-policy` to prevent OOM errors
- ☐ **Monitoring:** Collect Redis metrics (e.g., `redis_exporter` for Prometheus)
- ☐ **Cluster Mode:** Consider Redis Cluster or sharding for high throughput/HA
- ☐ **Backup:** Include Redis data in overall backup strategy

? **Related Stack:** [Database Foundation Stack ->](#)

MinIO (S3-Compatible Object Storage)

- ☐ **Security:** Use strong access keys, rotate credentials regularly
- ☐ **Erasure Coding:** Configure erasure coding for data redundancy and protection
- ☐ **Lifecycle Management:** Implement object lifecycle policies (e.g., delete old backups)
- ☐ **Versioning:** Enable object versioning to protect against accidental deletion
- ☐ **Replication:** Configure replication to a remote MinIO instance or S3 for disaster recovery
- ☐ **Monitoring:** Collect MinIO metrics (e.g., `minio_exporter` for Prometheus)
- ☐ **Disk Usage:** Monitor disk usage and alert before storage fills up

? **Related Stack:** [MinIO S3 Backup Stack ->](#)

? Workflow Automation (n8n)

- ☐ **Persistence:** Use PostgreSQL for workflow state, credentials, and execution history
- ☐ **Queue Mode:** Configure Redis for webhook and execution queue processing (separate worker nodes)
- ☐ **Execution Pruning:** Enable and configure execution data pruning to prevent database bloat
- ☐ **Security:** Limit n8n exposure, use secure credentials, protect webhook URLs

- ❑ **Monitoring:** Collect n8n metrics (e.g., execution count, failure rate) for observability
- ❑ **Backups:** Include n8n database and configuration in overall backup strategy
- ❑ **Error Handling:** Implement robust error handling and retry logic within workflows
- ❑ **Notifications:** Set up notifications for workflow failures (Slack, email, PagerDuty)

? **Related Stack:** [n8n Production Stack ->](#)

? Observability (Prometheus, Grafana, Loki)

- ❑ **Prometheus:** Configure Prometheus to scrape metrics from all services (node_exporter, cadvisor, app exporters)
- ❑ **Grafana:** Set up Grafana with dashboards for all components (node, containers, databases, n8n)
- ❑ **Loki:** Configure Loki/Promtail/Alloy for centralized log aggregation from all containers
- ❑ **Alerting:** Define Prometheus Alertmanager rules for critical metrics (CPU, memory, disk, service health)
- ❑ **Integrations:** Integrate Alertmanager with notification channels (Slack, PagerDuty, email)
- ❑ **Historical Data:** Configure long-term storage/retention for metrics and logs (e.g., Thanos, Mimir, Loki)
- ❑ **Access Control:** Secure Grafana with authentication (OAuth/GitHub/Keycloak) and RBAC
- ❑ **Dashboard Versioning:** Version control Grafana dashboards (JSON files in Git)

? **Related Stack:** [Observability Stack ->](#)

? Final Review

- ❑ **Documentation:** Document your entire infrastructure setup in Git (READMEs, configuration files)
 - ❑ **Testing:** Regularly test backups, disaster recovery procedures, and failover mechanisms
 - ❑ **Security Audit:** Perform periodic security audits and vulnerability scans
 - ❑ **Capacity Planning:** Monitor resource usage and plan for future scaling
 - ❑ **Automated Deployments:** Use CI/CD and Infrastructure as Code for all deployments (GitOps)
 - ❑ **Alerting Test:** Trigger and test all critical alerts (e.g., fill disk, stop database)
 - ❑ **Cost Monitoring:** Monitor cloud/hosting costs (if applicable) to avoid surprises
 - ❑ **Incident Response Plan:** Develop and practice an incident response plan
-

? Ready to Deploy?

Our **production stacks** are battle-tested, fully documented, and ready for your server:

Stack	Components	Use Case	Price
SSL Reverse Proxy Stack	Nginx + Certbot + CrowdSec	Secure entry point	EUR9.99
Database Foundation Stack	PostgreSQL + Redis	Data layer	EUR9.99
MinIO S3 Backup Stack	MinIO with lifecycle	Object storage	EUR9.99
n8n Production Stack	n8n + Postgres + Redis	Workflow automation	EUR9.99
Observability Stack	Prometheus + Grafana + Loki	Monitoring	EUR9.99

Stack	Components	Use Case	Price
[? Complete DevOps Bundle -> Save 25%]	ALL 5 stacks	Full production environment	EUR37.49 (save EUR12.46)

? Browse All Production Stacks -> <https://tobias-weiss.org/store>

? Need Help?

- **Questions about infrastructure?** Email: devops@tobias-weiss.org
- **Need 1:1 guidance for your deployment?** [Book a Production Stack Consultation ->](#)
- **Found a mistake?** Open an issue: github.com/tobias-weiss-ai-xr/next-tobias-weiss-org

Remember: Production readiness is a journey, not a destination. This checklist helps you get there faster.

(C) 2026 Tobias Weiss | tobias-weiss.org | All rights reserved

This checklist is provided as-is. No warranties express or implied. Infrastructure practices may change over time. Last updated: July 2026.

tobias-weiss.org

DevOps Engineering & Linux Certification

More Resources

- tobias-weiss.org Articles, guides, and free resources
 - courses.graphwiz.ai LPI practice exams & study tools
 - graphwiz.ai AI infrastructure & knowledge graphs
 - ki-kompetenz-training.de EU AI Act compliant AI literacy training
-

© 2025 Tobias Weiss. All rights reserved.

This document is provided as-is. No warranties express or implied.

Generated: <https://tobias-weiss.org>