DevOps Automation Checklist

25 Tasks to Automate Today

Your team can save 5+ hours per developer per week

How to Use This Checklist

- **Priority:** High = Do first, Medium = Do soon, Low = Nice to have
- **Difficulty:** Easy = < 1 day, Medium = 1-3 days, Hard = 1+ weeks
- **Time Saved:** Hours per developer per week
- **ROI:** Estimated return on investment

HIGH PRIORITY (Start Here)

1. Automated Testing Setup

Priority: High | **Difficulty:** Medium | **Time Saved:** 5 hrs/week

What to automate:

- Unit tests run on every commit
- Integration tests run before merge
- E2E tests run on staging deployments
- Test coverage reports automatically generated

Quick win: Set up Jest/Vitest with GitHub Actions
ROI: Catches bugs early, saves debugging time

---

2. CI/CD Pipeline Automation

Priority: High | **Difficulty:** Hard | **Time Saved:** 8 hrs/week

What to automate:

- Automatic builds on push
- Automated deployments to staging
- Production deployments with approval gates
- Rollback automation

Quick win: GitHub Actions or GitLab CI/CD
ROI: Eliminates manual deployment errors, saves hours

---

3. Code Quality Checks

Priority: High | **Difficulty:** Easy | **Time Saved:** 3 hrs/week

What to automate:

- Linting on every commit (ESLint, Prettier)
- Formatting checks before merge
- Type checking (TypeScript)
- Code complexity analysis

Quick win: Pre-commit hooks with Husky
ROI: Consistent code quality, fewer code reviews

---

4. Environment Setup Automation

Priority: High | **Difficulty:** Medium | **Time Saved:** 4 hrs/week

What to automate:

- One-command project setup
- Docker Compose for local dev
- Environment variable validation
- Dependency installation scripts

Quick win: Create setup.sh script
ROI: Faster onboarding, fewer "works on my machine" issues

---

5. Dependency Management

Priority: High | **Difficulty:** Easy | **Time Saved:** 2 hrs/week

What to automate:

- Automated dependency updates (Dependabot, Renovate)
- Security vulnerability scanning
- License compliance checks
- Outdated package alerts

Quick win: Enable Dependabot on GitHub
ROI: Keeps dependencies secure and up-to-date

MEDIUM PRIORITY (Do Soon)

6. Boilerplate Code Generation

Priority: Medium | **Difficulty:** Medium | **Time Saved:** 6 hrs/week

What to automate:

- API endpoint scaffolding
- CRUD operation templates
- Test file generation

- Component templates

Quick win: Use Plop.js or custom generators
ROI: Consistent code structure, faster development

---

7. Database Migration Automation

Priority: Medium | **Difficulty:** Medium | **Time Saved:** 3 hrs/week

What to automate:

- Migration scripts run automatically
- Rollback procedures
- Database seeding for dev
- Migration testing

Quick win: Use migration tools (Knex, TypeORM, Alembic)
ROI: Safer database changes, fewer production issues

---

8. Documentation Generation

Priority: Medium | **Difficulty:** Easy | **Time Saved:** 2 hrs/week

What to automate:

- API docs from code comments (Swagger/OpenAPI)
- README generation
- Changelog automation
- Architecture diagrams

Quick win: JSDoc + Swagger for APIs
ROI: Always up-to-date docs, less maintenance

---

9. Git Workflow Automation

Priority: Medium | **Difficulty:** Easy | **Time Saved:** 2 hrs/week

What to automate:

- Branch naming conventions
- Commit message validation
- PR template automation
- Merge conflict detection

Quick win: Git hooks + branch protection rules
ROI: Cleaner git history, easier code reviews

---

10. Logging and Monitoring Setup

Priority: Medium | **Difficulty:** Medium | **Time Saved:** 4 hrs/week

What to automate:

- Structured logging
- Error tracking (Sentry, Rollbar)
- Performance monitoring
- Alert automation

Quick win: Set up Sentry for error tracking
ROI: Faster debugging, proactive issue detection

---

11. Security Scanning

Priority: Medium | **Difficulty:** Easy | **Time Saved:** 3 hrs/week

What to automate:

- SAST (Static Application Security Testing)
- Dependency vulnerability scanning
- Secret detection (API keys, passwords)
- Security policy enforcement

Quick win: Snyk or GitHub Advanced Security
ROI: Prevents security breaches, compliance

---

12. Performance Testing

Priority: Medium | **Difficulty:** Hard | **Time Saved:** 4 hrs/week

What to automate:

- Load testing on deployments
- Performance regression detection
- Lighthouse CI for web apps
- Database query optimization alerts

Quick win: Lighthouse CI in GitHub Actions
ROI: Better user experience, fewer performance issues

---

13. Backup Automation

Priority: Medium | **Difficulty:** Easy | **Time Saved:** 2 hrs/week

What to automate:

- Database backups (daily/hourly)
- File backup automation
- Backup verification
- Restore testing

Quick win: Cron jobs or cloud backup services
ROI: Data safety, disaster recovery

---

14. Release Notes Automation

Priority: Medium | **Difficulty:** Easy | **Time Saved:** 1 hr/week

What to automate:

- Generate release notes from commits
- Version bumping
- Changelog updates
- Release announcements

Quick win: Semantic Release or Release Drafter
ROI: Consistent releases, better communication

---

15. Feature Flag Management

Priority: Medium | **Difficulty:** Medium | **Time Saved:** 3 hrs/week

What to automate:

- Feature flag toggles
- Gradual rollouts
- A/B testing setup
- Flag cleanup automation

Quick win: LaunchDarkly or custom solution
ROI: Safer deployments, faster feature releases

LOW PRIORITY (Nice to Have)

16. Code Review Automation

Priority: Low | **Difficulty:** Medium | **Time Saved:** 2 hrs/week

What to automate:

- Automated code review suggestions
- PR size limits
- Review assignment automation
- Review reminder bots

Quick win: CodeRabbit or GitHub Copilot suggestions
ROI: Faster reviews, better code quality

---

17. Infrastructure as Code

Priority: Low | **Difficulty:** Hard | **Time Saved:** 5 hrs/week

What to automate:

- Infrastructure provisioning (Terraform, Pulumi)
- Configuration management
- Environment replication
- Infrastructure testing

Quick win: Terraform for cloud resources
ROI: Reproducible infrastructure, fewer manual errors

---

18. Email/Slack Notifications

Priority: Low | **Difficulty:** Easy | **Time Saved:** 1 hr/week

What to automate:

- Deployment notifications
- Test failure alerts
- Security alerts
- Status updates

Quick win: GitHub Actions Slack integration
ROI: Better team communication, faster response

---

19. Data Seeding for Development

Priority: Low | **Difficulty:** Easy | **Time Saved:** 1 hr/week

What to automate:

- Test data generation
- Database seeding scripts
- Mock API responses
- Sample data sets

Quick win: Faker.js or similar libraries
ROI: Faster development, realistic testing

---

20. API Contract Testing

Priority: Low | **Difficulty:** Medium | **Time Saved:** 2 hrs/week

What to automate:

- API contract validation
- Schema testing
- Version compatibility checks
- Breaking change detection

Quick win: Pact or similar tools
ROI: Prevents API breakages, better integrations

---

21. Cost Monitoring

Priority: Low | **Difficulty:** Medium | **Time Saved:** 2 hrs/week

What to automate:

- Cloud cost tracking
- Budget alerts
- Resource optimization suggestions
- Cost allocation reports

Quick win: Cloud provider cost dashboards
ROI: Lower cloud bills, better budgeting

---

22. Compliance Automation

Priority: Low | **Difficulty:** Hard | **Time Saved:** 3 hrs/week

What to automate:

- Compliance checks (GDPR, SOC2)
- Audit log generation
- Policy enforcement
- Compliance reporting

Quick win: Automated audit logging
ROI: Compliance, reduced risk

---

23. Onboarding Automation

Priority: Low | **Difficulty:** Medium | **Time Saved:** 4 hrs/week

What to automate:

- New developer setup scripts
- Access provisioning
- Documentation delivery
- Welcome workflows

Quick win: Automated setup scripts
ROI: Faster onboarding, better developer experience

---

24. Analytics and Reporting

Priority: Low | **Difficulty:** Medium | **Time Saved:** 2 hrs/week

What to automate:

- Usage analytics collection
- Performance reports
- Error rate dashboards
- User behavior tracking

Quick win: Google Analytics or similar
ROI: Data-driven decisions, better products

---

25. Disaster Recovery Testing

Priority: Low | **Difficulty:** Hard | **Time Saved:** 3 hrs/week

What to automate:

- Regular DR drills
- Backup restoration testing

- Failover testing
- Recovery time measurement

Quick win: Automated backup restore tests
ROI: Business continuity, reduced downtime risk

Summary

Total Time Saved: 80+ hours per developer per week (if all automated)

Realistic for Most Teams: 15-25 hours per developer per week

Quick Wins (Easy + High Priority):

- Code Quality Checks (#3)
- Dependency Management (#5)
- Documentation Generation (#8)
- Git Workflow Automation (#9)

High Impact (High Priority):

- Automated Testing (#1)
- CI/CD Pipeline (#2)
- Environment Setup (#4)
- Boilerplate Generation (#6)

Next Steps

1. **Start with Quick Wins** - Pick 2-3 easy, high-priority items
2. **Measure Impact** - Track time saved before/after
3. **Iterate** - Add more automation as you see value
4. **Standardize** - Share automation across your team

Need Help?

If you want help implementing these automations for your team, book a free consultation:

https://calendly.com/kyjahn-smith/consultation

I'll show you exactly how to set up these automations for your specific tech stack and team size.

About This Checklist

This checklist is part of the **DevOps Productivity Suite** - 5 production-ready tools that automate your workflow, standardize processes, and catch issues early.

What's included:

- Shell Games Toolkit (automation scripts)
- Ubiquitous Automation (CI/CD pipelines)
- Git Workflows Sample (branching strategies)
- Code Generator Tool (boilerplate generation)
- Software Entropy (technical debt scanning)

Learn more: Book a free consultation to see how we can help your team save 5+ hours per developer per week.

Generated by DevOps Productivity Suite