Lightning-Fast Developer Onboarding Checklist

This checklist is designed to make onboarding new developers across nearly any environment as smooth, fast, and automated as possible.

---

1. Pre-Onboarding (Before Day One)

- ☐ Ensure company-issued hardware is prepared (or remote access configured)
- ☐ Grant access to email, internal chat (Slack, Teams), and calendar
- ☐ Provide access to password manager (e.g. Bitwarden, 1Password)
- ☐ Share onboarding documentation link (README, handbook)
- ☐ Assign onboarding buddy or technical mentor
- ☐ Invite to source code repositories (GitHub, GitLab, Bitbucket)
- ☐ Grant access to CI/CD pipelines
- ☐ Grant access to internal dashboards, monitoring tools (Grafana, Kibana)
- ☐ Pre-provision cloud accounts (AWS, Azure, GCP)
- ☐ Schedule intro meetings with team leads, PMs, and other devs
- ☐ Add to mailing lists, documentation platforms (Confluence, Notion)
- ☐ Provide SSO or identity provider (Okta, Auth0) credentials

2. Machine Setup

- ☐ Install operating system (if needed) or use pre-configured image
- ☐ Install package manager (Homebrew, Chocolatey, apt, dnf)
- ☐ Install essential CLI tools (git, curl, wget, jq, etc.)

- ☐ Install language runtimes (e.g. Node.js, Python, Java, Ruby, Go, Rust)
- ☐ Install language version manager (nvm, pyenv, rbenv, sdkman)
- ☐ Install container tooling (Docker, Podman)
- ☐ Install devcontainer support (Docker Desktop + VS Code extension)
- ☐ Install code editor/IDE (VS Code, IntelliJ, Vim, Emacs)
- ☐ Clone repository using a single command/script
- ☐ Configure `.gitconfig` and SSH keys
- ☐ Connect to VPN if needed

3. Local Dev Environment Setup

- ☐ One-liner `make setup` or `task setup` command available
- ☐ Devcontainer setup available and documented
- ☐ Docker Compose or Kubernetes manifests included for local stack
- ☐ Scripts to decrypt secrets (SOPS, Vault CLI)
- ☐ `.env` files or environment injection available
- ☐ Secrets mounted or injected securely (dotenv, vault-agent)
- ☐ Editor settings and workspace configuration provided
- ☐ Database migrations pre-seeded or mock data available
- ☐ Option to use remote containers (e.g. Codespaces, Gitpod)

4. Project Familiarization

- ☐ High-level architecture diagram provided
- ☐ Tech stack explained (services, frameworks, tools)
- ☐ Common workflows documented (e.g. how to run tests, lint, format)
- ☐ Links to API documentation (internal Swagger/OpenAPI docs)
- ☐ Runbook or service guide included
- ☐ Test data fixtures or dummy users pre-configured
- ☐ Access to staging/test environments

5. Development Workflow

- ☐ Branching strategy explained (Git Flow, trunk-based)
- ☐ Linting/formatting tools auto-run (e.g. pre-commit, husky)
- ☐ Unit/integration test setup explained
- ☐ Debugging guide provided
- ☐ Feature flag system explained
- ☐ Deployment pipeline overview (CI/CD stages)
- ☐ Static analysis tools available and documented
- ☐ Performance profiling/debugging tools listed

6. Communication and Support

- ☐ Team structure explained
- ☐ Who to ask for what (tech lead, infra, QA, etc.)
- ☐ List of helpful Slack channels

- [ ] Escalation path for blockers
- [ ] Working hours, expectations, and rituals explained (standups, retros)
- [ ] Tickets/issues assigned on Day One

7. Day-One Checklist

- [ ] Developer can run the app locally
- [ ] Developer can run the test suite
- [ ] Developer can deploy to staging (if needed)
- [ ] Developer can make a commit, push, and create a pull request
- [ ] Developer completes a "first task" or bug fix
- [ ] Developer attends standup or intro meeting
- [ ] Developer feedback collected after Day One

8. Ongoing Support

- [ ] Weekly check-in scheduled for first month
- [ ] Feedback loop on onboarding process
- [ ] Tickets evolve from minor tasks to core features over time
- [ ] Encourage documentation contributions as part of ramp-up
- [ ] Performance review milestone scheduled

9. Optional but Helpful

- [ ] Access to design tools (Figma, Sketch)
- [ ] Access to product roadmap or backlogs (Jira, Trello)
- [ ] Internal API gateways or Postman collections shared
- [ ] Feature toggles explained and accessible
- [ ] Shadowing opportunity with senior developers
- [ ] Invite to team rituals (demo day, team lunch, hackathons)

Credits go to Tom Herder of Byteherder