GitHub Access Policy

Document ID: IT-SW-003 **Last Updated:** 2024-02-05 **Owner:** Platform Engineering **Classification:** Internal

---

Overview

NovaTech uses GitHub Enterprise for source code management. This policy covers access levels, security requirements, and best practices for GitHub usage.

Organization Structure

NovaTech GitHub Organization

- **URL:** https://github.com/novatech
- **Plan:** GitHub Enterprise Cloud
- **SSO:** Required via Okta

Teams Structure

Team	Repositories	Members
@novatech/platform	platform-, *infra-*	Platform engineers
@novatech/cloudforge	cloudforge-*	CloudForge team
@novatech/devpipeline	devpipeline-*	DevPipeline team
@novatech/securevault	securevault-*	SecureVault team
@novatech/datalens	datalens-*	DataLens team
@novatech/security	All (read)	Security team

Access Levels

Repository Permissions

Level	Capabilities
Read	Clone, view code, view issues
Triage	Read + manage issues/PRs
Write	Triage + push, merge PRs
Maintain	Write + manage repo settings
Admin	Full control (limited distribution)

Default Access

- **New engineers:** Read access to all public internal repos
- **Team membership:** Write access to team repositories
- **Cross-team:** Request via team lead

Account Requirements

Account Setup

1. Use your NovaTech email for your GitHub account
2. If you have a personal GitHub account, link your NovaTech email
3. Enable SSO: Visit github.com/orgs/novatech and authorize via Okta

Security Requirements

- **2FA:** Required (enforced by organization)
- **SSO:** Required for organization access
- **SSH Keys:** Required for Git operations (no HTTPS passwords)

SSH Key Setup

```
# Generate key
ssh-keygen -t ed25519 -C "your.name@novatech.com"

# Add to GitHub
cat ~/.ssh/id_ed25519.pub
# Copy output to GitHub > Settings > SSH Keys
```

Repository Policies

Branch Protection

All repositories must have: - Protected `main` branch - Required PR reviews (minimum 1) - Required status checks passing - No force pushes to main

Code Review Requirements

Repository Type	Required Reviewers
Product code	1 team member
Infrastructure	1 team + 1 platform
Security-sensitive	1 team + 1 security

Merge Strategies

- **Squash merge:** Default for feature branches
- **Merge commit:** For release branches
- **Rebase:** Developer preference for local work

Security Practices

Secrets Management

Never commit secrets to repositories: - API keys - Passwords - Private keys - Connection strings

Use: - GitHub Secrets for Actions - AWS Secrets Manager for runtime - .env.example for templates (never .env)

Secret Scanning

- Enabled on all repositories
- Push protection blocks known secret patterns
- If secret detected: rotate immediately, notify security@novatech.com

Dependency Scanning

- Dependabot enabled on all repositories
- Security alerts reviewed weekly
- Critical vulnerabilities patched within 48 hours

GitHub Actions

Self-Hosted Runners

Available for: - Long-running tests - GPU workloads - Internal network access

Request via Platform team.

Action Allowlist

Only GitHub-verified and NovaTech-approved actions permitted: - `actions/*` (official) - `novatech/*` (internal) - Pre-approved community actions (see wiki)

Secrets in Actions

- Use organization secrets for shared values
- Use repository secrets for repo-specific values
- Use environment secrets for deployment targets

Personal Access Tokens

Token Types

Type	Use Case	Expiration
Fine-grained	Automation, CI/CD	90 days max
Classic	Legacy (discouraged)	30 days max

Creating Tokens

1. Go to GitHub > Settings > Developer Settings > Personal Access Tokens
2. Select Fine-grained token
3. Set resource owner to “novatech”
4. Select minimal required permissions
5. Set expiration (max 90 days)

Token Security

- Never share tokens
- Never commit tokens to code
- Rotate tokens regularly
- Revoke unused tokens

Repository Naming Conventions

`<product>-<component>[-<type>]`

Examples: - `cloudforge-api` - `cloudforge-web` - `cloudforge-docs` - `devpipeline-core` - `platform-terraform-modules`

Creating New Repositories

Process

1. Submit request in #platform-requests
2. Provide:

    - Repository name (following conventions)
    - Description
    - Team ownership
    - Public/Private/Internal visibility

3. Platform team creates with standard settings

Standard Setup

All new repositories include: - Branch protection rules - CODEOWNERS file - Issue/PR templates - CI workflow template - README template

---

Offboarding

When leaving NovaTech: - GitHub SSO access revoked automatically - Personal account remains but org access removed - Open PRs reassigned - Transfer repository ownership if applicable

---

Support

- Slack: #platform-help
- Email: platform@novatech.com
- GitHub issues: novatech/platform-support

Related Documents: Development Environment Setup (IT-SW-002), Security Best Practices (IT-SEC-010), Code Review Guidelines (ENG-DEV-005)