Sección: Evidencia de ejecución del pipeline CI/CD DevSecOps
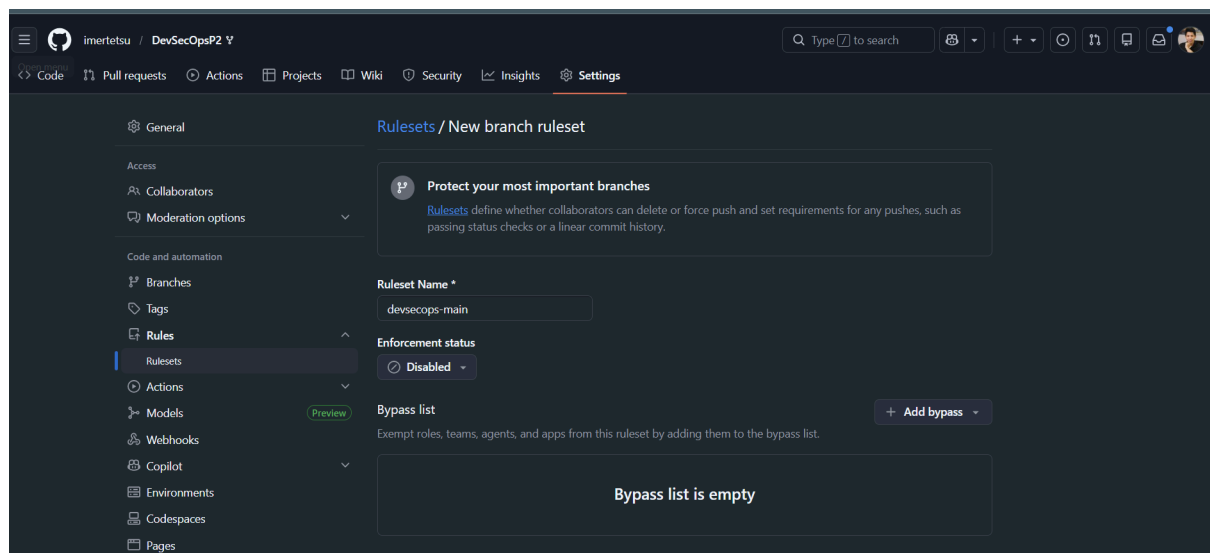
Objetivo de esta sección

Demostrar que el pipeline CI/CD implementado:

- se ejecuta correctamente en GitHub Actions
- aplica controles automáticos de calidad y seguridad
- bloquea el despliegue cuando hay errores (gates)
- permite avanzar solo cuando todo está correcto

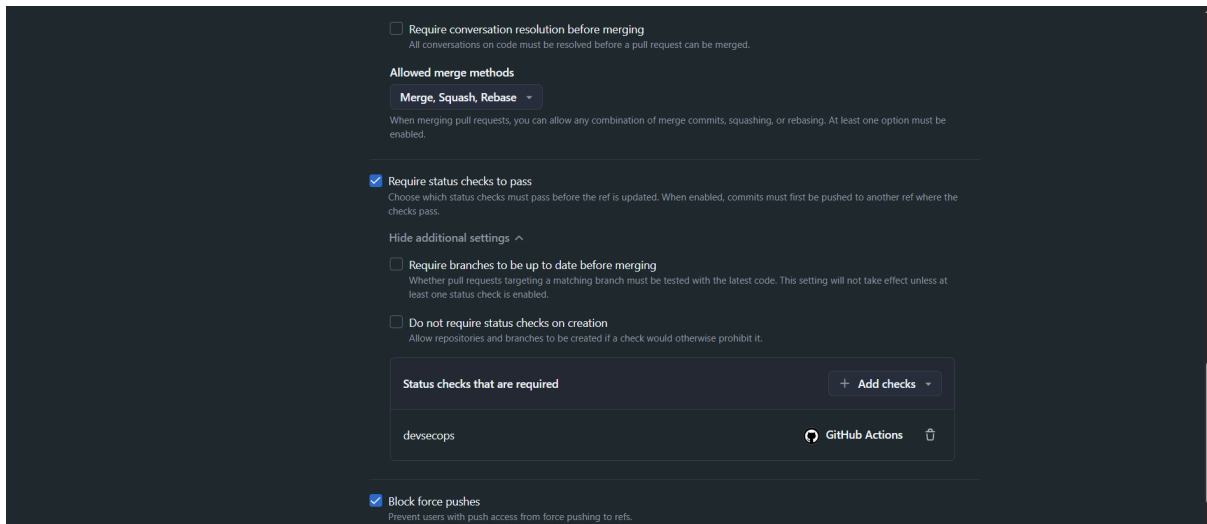Esto valida el enfoque DevSecOps exigido en la tarea.

Branch Protection Rules (Required Checks)

Configuración aplicada

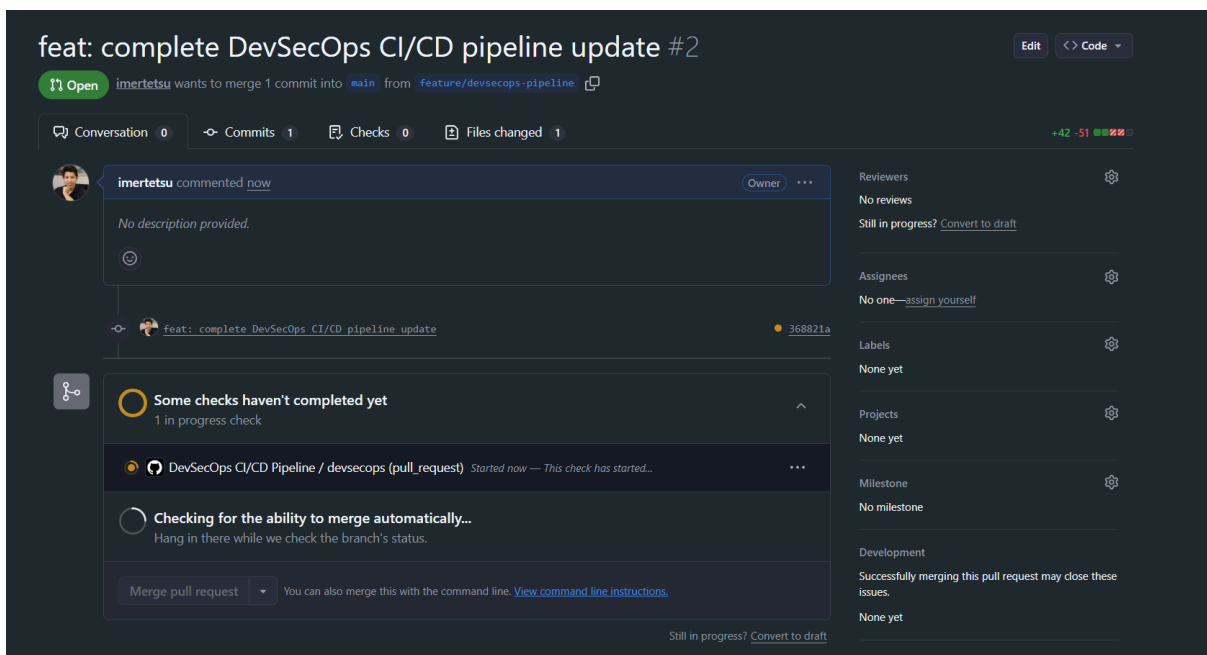Se configuró protección de la rama `main` con:

- Require pull request before merging
- Require status checks to pass
- Required check: `devsecops` workflow

Esto garantiza que:

- nadie puede hacer push directo a `main`
- todo cambio debe pasar por el pipeline
- el pipeline actúa como **mecanismo obligatorio de control**

Ejecución fallida del pipeline (gate de calidad/seguridad)

Se ejecutó el pipeline con un commit que contiene errores de ESLint

Resultado observado:
El pipeline falla en la etapa de Code Quality y detiene la ejecución.

Ejecución exitosa del pipeline

Se corrigieron los errores de lint/seguridad y se ejecutó nuevamente el pipeline.

Resultado observado: El pipeline completa todas las etapas exitosamente:

- Install
- Lint
- Tests
- SAST (Semgrep)
- SCA (npm audit)
- Docker build
- Trivy scan
- Smoke test / healthcheck

Link a ejecución en GitHub Actions:
https://github.com/imertetsu/DevSecOpsP2/actions/runs/22279636991

Conclusión de la evidencia

Las evidencias demuestran que el pipeline implementado:

- ejecuta controles automáticos de calidad, testing y seguridad
- bloquea integraciones defectuosas
- valida el despliegue con pruebas de salud
- impone políticas de protección en la rama principal

Por lo tanto, el sistema cumple con un flujo CI/CD DevSecOps funcional y seguro, alineado con los requerimientos de la tarea.