

GitHub Migration Plan

Phase 1: Pre-Migration Assessment and Preparation

1. Artifactory Assessment
 - Objective: Review the current usage of Artifactory and identify any dependencies or artifacts that may need migration or integration with GitHub.
 - Action Steps:
 - Audit Artifactory repositories used by teams.
 - Document required artifacts and dependencies for the migration.
 - Plan for necessary integrations with GitHub Actions or workflows if Artifactory is used in CI/CD pipelines.
2. Branching Protection Review
 - Objective: Ensure secure and consistent branching strategies across all repositories to avoid any disruption during or post-migration.
 - Action Steps:
 - Review the current branch protection rules and permissions for critical branches.
 - Document and implement consistent branching rules (e.g., enforce pull requests, require code reviews).
 - Communicate branching policies to all team members.
3. User and Access Review
 - Objective: Review user access to repositories and set permissions based on roles to ensure a secure migration.
 - Action Steps:
 - Audit user roles and permissions for all repositories.
 - Revoke or adjust permissions as necessary based on users' current roles.
 - Implement Single Sign-On (SSO) if available and enforce two-factor authentication (2FA).

Phase 2: Repository Preparation and Security Review

4. Existing Repo Clean Up
 - Objective: Review and clean up existing repositories to streamline the migration process.
 - Action Steps:
 - Archive or delete unused repositories.
 - Remove deprecated branches and unmaintained code.
 - Rename repositories if needed to maintain a consistent naming convention.
5. Remove any GitHub Personal Access Tokens (PAT) or AWS Access Keys/Secrets from repos
 - Objective: Ensure there are no sensitive credentials stored in repositories to enhance security.
 - Action Steps:

- Use tools like GitHub Secret Scanning to identify and remove PATs, AWS access keys, or other sensitive information.
 - Replace any required credentials with GitHub secrets, environment variables, or other secure storage solutions.
 - Communicate this process to teams and provide guidelines on securely managing credentials.
6. Internal and External Dependencies Documented
- Objective: Document all dependencies to ensure smooth integration and dependency management post-migration.
 - Action Steps:
 - List internal and external dependencies used in each repository.
 - Update dependencies to compatible versions if necessary.
 - Include dependency documentation in each repository or project's README file.

Phase 3: Migration Execution and Testing

7. Reserve one sprint - two weeks - prior to December 16, 2024, if not already scheduled
- Objective: Dedicate a sprint solely for the migration, to minimize impact on other work and allow for thorough testing.
 - Action Steps:
 - Schedule the sprint from December 2-13, 2024.
 - Notify stakeholders of limited availability during this time for any non-critical tasks.
 - Allocate team resources to perform the migration and to address any immediate issues.
8. Submit GEV ES GitHub Migration Tracker Form - if not already performed
- Objective: Ensure all required forms and documentation are completed for audit and compliance purposes.
 - Action Steps:
 - Complete and submit the GEV ES GitHub Migration Tracker Form.
 - Verify that any pre-migration requirements, such as security or compliance checklists, are met.
 - Maintain a copy of the form and any related documents for reference.

Phase 4: Post-Migration Validation and Monitoring

9. Post-Migration Review
- Objective: Ensure the migration was successful and that repositories are functioning correctly in the GitHub environment.
 - Action Steps:
 - Perform a thorough test of all CI/CD pipelines, integrations, and dependencies.
 - Address any issues or errors that arise during testing.

- Monitor repository access, permissions, and activity logs to verify security configurations.

10. Documentation and Knowledge Sharing

- Objective: Document the migration process and provide training for any new workflows or policies.

- Action Steps:
 - Update internal documentation to reflect new GitHub workflows and access policies.
 - Hold training sessions for team members to cover new processes and security practices.
 - Collect feedback from teams to identify areas for further improvement.

Timeline Overview

- Phase 1: Pre-Migration Assessment and Preparation - November 1-15, 2024
- Phase 2: Repository Preparation and Security Review - November 16-30, 2024
- Phase 3: Migration Execution and Testing - December 2-13, 2024
- Phase 4: Post-Migration Validation and Monitoring - December 16-20, 2024