OKLAHOMA
Office of Management
& Enterprise Services

Code Escrow Standard

Introduction

The Code Escrow process was developed to pull contractor-developed code into the State's GitHub enterprise without the need to assign licenses to contractors. The contractor's code changes will be pulled on a regular schedule and a manual pull request review is required before any changes can be merged.

Purpose

This standard outlines the code escrow process that is to be followed when code is developed by contractors outside of Oklahoma's enterprise GitHub organizations. The process is facilitated by an Azure DevOps pipeline.

Standard

- One-time setup that needs to be performed once for each organization that hosts the code escrow YAML.
    - Clone the repository with the code escrow YAML and publish it to the enterprise organization that will facilitate the process. If you do not have access to the ok-osdh-escrow organization, please submit a request to the OMES Service Desk.
        - https://github.com/ok-osdh-escrow/workflows
    - Create a GitHub Personal Access Token.
        - Give the token a scope of repo.
        - Authorize the token for the organization that hosts the code escrow YAML.
    - Create a pipeline in Azure DevOps using the YAML in the Code Escrow repository.
        - Create a pipeline variable named GH_TOKEN and set the value of the variable to the token created in the previous step.
        - Make the variable a secret.
- Ask the contractor to provide an HTTPS link that includes a token or password which will be used to clone the source repository.
    - When choosing an expiration date for the token or password, consider how long development is expected to continue. If the token expires and development is still in progress, a new token or password will need to be generated and used.
    - If the contractor's code is stored in GitHub, ask for a link including a Personal Access Token with all repo permissions. Suggest that the contractor create a new account that only has access to the necessary repositories.
    - If the code is stored in GitLab, ask for a link including a deploy token with a scope of "read_repository".
    - If the code is stored in Bitbucket, ask for a link including the username and app password. Suggest that the contractor create a new account that only has access to the necessary repositories.
- Create a private internal repository in GitHub for each of the contractor repositories that will be cloned for the escrow process.
    - Get the HTTPS link used to clone the destination repository and save it for a later step.

- Create a Personal Access Token in GitHub with the repo scope and an expiration date that exceeds the expected duration of development for this code. Authorize the token for the organization.
    - Add the PAT to the HTTPS link obtained in the previous step so it can be used in the next step.
- Open the Azure DevOps pipeline created in the one-time setup step and edit it. Click on Variables to modify pipeline variables.
    - Add two new pipeline variables for each repository to be cloned. One variable will contain the HTTPS link provided by the contractor. The second variable will contain the HTTPS link with the PAT generated in the previous step.
    - Add a set of parameters for each repository that will be cloned. Repo parameters consist of name, source, sourceBranch, destination and cloneFolder.
        - name – Short, descriptive name for the repository.
        - source – Reference the pipeline variable for the source repository.
        - sourceBranch – Name of the branch to be pulled from the source repository.
        - destination – Reference the pipeline variable for the destination repository.
        - cloneFolder – Name of the folder that the source repository will be cloned into.
- Run the pipeline manually to verify everything is working as expected.
- After the first successful run turn on branch protection rules and add a CODEOWNERS file to the repository.

Compliance

This standard shall take effect upon publication and is made pursuant to Title 62 O.S. §§ 34.11.1 and 34.12 and Title 62 O.S. § 35.8. OMES IS may amend and publish the amended standards policies and standards at any time. Compliance is expected with all published policies and standards, and any published amendments thereof. Employees found in violation of this standard may be subject to disciplinary action, up to and including termination.

References

- GitHub CODEOWNERS File
- GitHub – Create Personal Access Token
- Azure DevOps Pipeline Variables
- Azure DevOps Cron Syntax
- GitLab Deploy Tokens
- Bitbucket App Passwords

Revision history

This standard is subject to periodic review to ensure relevancy.

Effective date: 03/30/2023	**Review Cycle:** Annual
Last revised: 1/31/2023	**Last reviewed:** 07/14/2023
Approved by: Joe McIntosh, Chief Information Officer	