Phase 3: Pilot programs

In this article

You may benefit from beginning with a few high-impact projects and teams with which to pilot an initial rollout. This will allow an initial group within your company to get familiar with GHAS, learn how to enable and configure GHAS, and build a solid foundation on GHAS before rolling out to the remainder of your company.

This article is part of a series on adopting GitHub Advanced Security at scale. For the previous article in this series, see "Phase 2: Preparing to enable at scale."

About pilot programs

We recommend you identify a few high-impact projects or teams to use in a pilot rollout of GHAS. This allows an initial group within your company to get familiar with GHAS and builds a solid foundation for GHAS before you roll it out to the remainder of your company.

The steps in this phase will help you enable GHAS on your enterprise, begin using its features, and review your results. If you're working with GitHub Professional Services, they can provide additional assistance through this process through onboarding sessions, GHAS workshops, and troubleshooting as needed.

Before you start your pilot projects, we recommend that you schedule some meetings for your teams, such as an initial meeting, midpoint review, and a wrap-up session when the pilot is complete. These meetings will help you all make adjustments as needed and ensure your teams are prepared and supported to complete the pilot successfully.

You need to enable GHAS for each pilot project, either by enabling the GHAS features for each repository or for all repositories in any organizations taking part in the pilot. For more information, see "Managing security and analysis settings for your repository" or "Managing security and analysis settings for your organization"

Piloting code scanning

You can run code scanning on a repository by creating a GitHub Actions workflow to run the CodeQL action. Code scanning uses GitHub-hosted runners by default, but this can be customized if you plan to host your own runner with your own hardware specifications. For more information, see "Hosting your own runners."

For more information about GitHub Actions, see:

- "Learn GitHub Actions"
- "Understanding GitHub Actions"
- "Events that trigger workflows"
- "Workflow syntax for GitHub Actions"

We recommend enabling code scanning on a repository-by-repository basis as part of your pilot program. For more information, see "Configuring advanced setup for code scanning."

If you want to enable code scanning for many repositories, you may want to script the process.

For an example of a script that opens pull requests to add a GitHub Actions workflow to multiple repositories, see the `jhutchings1/Create-ActionsPRs` repository for an example using PowerShell, or `nickliffen/ghas-enablement` for teams who do not have PowerShell and instead would like to use NodeJS.

When running initial code scans, you may find that no results are found or that an unusual number of results are returned. You may want to adjust what is flagged in future scans. For more information, see "Customizing your advanced setup for code scanning."

If your company wants to use other third-party code analysis tools with GitHub code scanning, you can use actions to run those tools within GitHub. Alternatively, you can upload results, which are generated by third-party tools as SARIF files, to code scanning. For more information, see "Integrating with code scanning."

Piloting secret scanning

GitHub scans repositories for known types of secrets, to prevent fraudulent use of secrets that were committed accidentally.

You need to enable secret scanning for each pilot project, either by enabling the feature for each repository or for all repositories in any organizations taking part in the project. For more information, see "Managing security and analysis settings for your repository" or "Managing security and analysis settings for your organization."

If you have collated any custom patterns specific to your enterprise, especially any related to the projects piloting secret scanning, you can configure those. For more information, see "Defining custom patterns for secret scanning."

To learn how to view and close alerts for secrets checked into your repository, see "Managing alerts from secret scanning."

For the next article in this series, see "Phase 4: Create internal documentation."