

Migrating from Bamboo with GitHub Actions Importer

Prerequisites

- **Bamboo Account:** You need access to a Bamboo account or organization with projects and pipelines that you want to migrate.
- **Bamboo Version:** Ensure your Bamboo version is 7.1.1 or greater.
- **Bamboo Personal Access Token:** Obtain a Bamboo personal access token for your account or organization. This token will be used to authenticate with Bamboo during the migration process.
- **Environment Setup:** You need an environment where you can run Linux-based containers and install necessary tools.
 - **Docker:** Ensure Docker is installed and running.
 - **GitHub CLI:** Install GitHub CLI.

Limitations

- **YAML Specification:** GitHub Actions Importer relies on the YAML specification generated by Bamboo Server. When Bamboo doesn't support exporting something to YAML, the missing information is not migrated.
- **Trigger Conditions:** Unsupported trigger conditions are surfaced as comments and transformed without them.
- **Artifacts:** Plans with customized artifact settings are not transformed. Instead, artifacts are stored and retrieved using the upload-artifact and download-artifact actions.
- **Disabled Plans, Jobs, and Tasks:** Disabled plans, jobs, and tasks may not be transformed or need manual adjustment.
- **Build Workspaces Cleanup:** Options for cleaning up build workspaces after a build is complete are not transformed.
- **Hanging Build Detection:** Hanging build detection options are not transformed.
- **Pattern Match Labeling:** Pattern match labeling is not transformed.
- **Permissions:** Permissions are not transformed because there is no suitable equivalent in GitHub Actions.
- **Project and Plan Variables:** If the Bamboo version is between 7.1.1 and 8.1.1, project and plan variables will not be migrated.

Installing the GitHub Actions Importer CLI Extension

- Open your terminal.
- Run the following command to install the GitHub Actions Importer CLI extension:

```
gh extension install github/gh-actions-importer
```

- After installation, verify that the extension is installed by running:

```
gh actions-importer -h
```

This command should display the available options and commands.

Installing Bamboo Docker and PostgreSQL

- Install Docker on your system if not already installed.
- Run the following commands to set up Bamboo Docker container:

```
docker network create bamboo
```

```
docker run --name bamboo-server --rm --detach --network bamboo --publish  
8085:8085 atlassian/bamboo-server
```

- Run the following commands to set up PostgreSQL Docker container (if not using an existing PostgreSQL instance):

```
docker run --name postgres-db --rm --detach --network bamboo --env  
POSTGRES_PASSWORD=mysecretpassword --env POSTGRES_USER=bamboo --env  
POSTGRES_DB=bamboo --publish 5432:5432 postgres
```

NOTES

Create GitHub Personal Access Token:

- Navigate to your GitHub account settings.
- Go to the "Developer settings" or "Developer options."
- Select "Personal access tokens."
- Generate a new token with the "workflow" scope.
- Copy the token and save it securely.

Create Bamboo Personal Access Token:

- Log in to your Bamboo instance.
- Access the "Personal Access Tokens" section.
- Generate a new token with the required permissions:
 - View, View Configuration, Edit for Build Plan
 - View, View Configuration, Edit for Deployment Project
 - View, View Configuration, Edit for Deployment Environment
- Copy the token and save it securely.

Perform an audit of Bamboo

To perform an audit of a Bamboo instance using GitHub Actions Importer, follow these steps:

Run the Audit Command:

- Open your terminal and execute the following command:

```
gh actions-importer audit bamboo --output-dir tmp/audit
```

This command will fetch all projects defined in your Bamboo organization, convert each pipeline to its equivalent GitHub Actions workflow, and generate a report summarizing the migration feasibility.

Inspect the Audit Results:

After the command completes, navigate to the specified output directory (tmp/audit) to find the audit report.

- Review the audit_summary.md file for a high-level summary of the audit results.
- Explore the sections within the summary, including Pipelines, Build Steps, Manual Tasks, and Files, to understand the details of the migration analysis.

Analyze Pipelines:

- Check the "Pipelines" section to understand the conversion rate and success of each pipeline migration.
- Identify successful, partially successful, unsupported, and failed pipelines based on the provided criteria.

Review Build Steps:

- Examine the "Build Steps" section to understand which build steps were automatically converted, unknown, or unsupported.
- Take note of any manual tasks required for unsupported build steps.

Address Manual Tasks:

- Check the "Manual Tasks" section to identify tasks that GitHub Actions Importer couldn't complete automatically.
- Handle tasks such as creating repository secrets or defining self-hosted runners manually.

Explore Files and Usage Data:

- Navigate to the files section to inspect original and converted pipeline files, network responses, and stack traces.
- Review the workflow_usage.csv file to understand the actions, secrets, and runners used by each successfully converted pipeline.

Forecasting usage

To forecast potential GitHub Actions usage from completed pipeline runs in your Bamboo instance, follow these steps:

Run the Forecast Command:

- Open your terminal and execute the following command to perform the forecast:

```
gh actions-importer forecast bamboo --output-dir tmp/forecast_reports
```

- This command will compute metrics from completed pipeline runs and generate a forecast report.

Limit Forecast to a Project (Optional):

- If you want to limit the forecast to the plans and deployment environments associated with a specific project, you can use the `--project` option followed by the project key.
- For example:

```
gh actions-importer forecast bamboo --project PAN --output-dir  
tmp/forecast_reports
```

Inspect the Forecast Report:

After the command completes, navigate to the specified output directory (`tmp/forecast_reports`) to find the forecast report (`forecast_report.md`).

- Open the report to review the forecasted metrics and insights.
- Look for key terms such as job count, pipeline count, execution time, queue time metrics, and concurrent jobs metrics.

Analyze Forecast Metrics:

- Review the job count and pipeline count to understand the total workload and complexity of the pipelines.
- Consider execution time and queue time metrics to plan for the cost of GitHub-hosted runners and optimize resource allocation.
- Analyze concurrent jobs metrics to determine the number of jobs running simultaneously and adjust runner configurations accordingly.

Perform a dry-run migration of a Bamboo pipeline

To perform a dry-run migration of a Bamboo pipeline to GitHub Actions, follow these steps:

Run Dry-Run Migration for a Build Plan:

- If you're migrating a Bamboo build plan, execute the following command in your terminal:

```
gh actions-importer dry-run bamboo build --plan-slug :my_plan_slug --output-  
dir tmp/dry-run
```

- Replace `:my_plan_slug` with the plan's project and plan key in the format `<projectKey>-<planKey>` (e.g., PAN-SCRIP).

Run Dry-Run Migration for a Deployment Project:

- If you're migrating a Bamboo deployment project, execute the following command:

```
gh actions-importer dry-run bamboo deployment --deployment-project-id
:my_deployment_project_id --output-dir tmp/dry-run
```

- Replace `:my_deployment_project_id` with the ID of the deployment project you are converting.

Inspect Dry-Run Results:

Navigate to the specified output directory (`tmp/dry-run`) to view the logs of the dry run and the converted workflow files.

- Check if GitHub Actions Importer was able to convert the pipeline automatically.
- Look for any unknown build steps or partially successful conversions.

Customize Conversion Process (Optional):

If GitHub Actions Importer couldn't convert certain aspects automatically, such as unknown build steps or partially successful pipelines, consider creating custom transformers to further customize the conversion process.

- Refer to the documentation for information on extending GitHub Actions Importer with custom transformers.

Perform a production migration of a Bamboo pipeline

To perform a production migration of a Bamboo pipeline to GitHub Actions, follow these steps:

Run Migration for a Build Plan:

- If you're migrating a Bamboo build plan, execute the following command in your terminal:

```
gh actions-importer migrate bamboo build --plan-slug :my_plan_slug --target-
url :target_url --output-dir tmp/migrate
```

- Replace `:my_plan_slug` with the plan's project and plan key in the format `<projectKey>-<planKey>`, and `:target_url` with the URL for your GitHub repository.

Review Migration Output:

- The command's output will include the URL to the pull request that adds the converted workflow to your repository.

Run Migration for a Deployment Project (Optional):

- If you're migrating a Bamboo deployment project, execute the following command:

```
gh actions-importer migrate bamboo deployment --deployment-project-id  
:my_deployment_project_id --target-url :target_url --output-dir tmp/migrate
```

- Replace `:my_deployment_project_id` with the ID of the deployment project you are converting, and `:target_url` with the URL for your GitHub repository.

Review Pull Request:

- Navigate to the provided URL to review the pull request that adds the converted workflow to your GitHub repository.