Using OpenID Connect with reusable workflows

In this article

You can use reusable workflows with OIDC to standardize and security harden your deployment steps.

About reusable workflows

Rather than copying and pasting deployment jobs from one workflow to another, you can create a reusable workflow that performs the deployment steps. A reusable workflow can be used by another workflow if it meets one of the access requirements described in "Reusing workflows."

You should be familiar with the concepts described in "Reusing workflows" and "About security hardening with OpenID Connect."

Defining the trust conditions

When combined with OpenID Connect (OIDC), reusable workflows let you enforce consistent deployments across your repository, organization, or enterprise. You can do this by defining trust conditions on cloud roles based on reusable workflows. The available options will vary depending on your cloud provider:

- **Using `job_workflow_ref`**:
  - To create trust conditions based on reusable workflows, your cloud provider must support custom claims for `job_workflow_ref`. This allows your cloud provider to identify which repository the job originally came from.
  - For clouds that only support the standard claims (audience (`aud`) and subject (`sub`)), you can use the API to customize the `sub` claim to include `job_workflow_ref`. For more information, see "About security hardening with OpenID Connect". Support for custom claims is currently available for Google Cloud Platform and HashiCorp Vault.
- **Customizing the token claims**:
  - You can configure more granular trust conditions by customizing the issuer (`iss`) and subject (`sub`) claims that are included with the JWT. For more information, see "About security hardening with OpenID Connect".

How the token works with reusable workflows

During a workflow run, GitHub's OIDC provider presents a OIDC token to the cloud provider which contains information about the job. If that job is part of a reusable workflow, the token will include the standard claims that contain information about the calling workflow, and will also include a custom claim called `job_workflow_ref` that contains information about the called workflow.

For example, the following OIDC token is for a job that was part of a called workflow. The `workflow`, `ref`, and other attributes describe the caller workflow, while `job_workflow_ref` refers to the called workflow:

YAML


```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "example-thumbprint",
  "kid": "example-key-id"
}
{
  "jti": "example-id",
  "sub": "repo:octo-org/octo-repo:environment:prod",
  "aud": "https://github.com/octo-org",
  "ref": "refs/heads/main",
  "sha": "example-sha",
  "repository": "octo-org/octo-repo",
  "repository_owner": "octo-org",
  "actor_id": "12",
  "repository_id": "74",
  "repository_owner_id": "65",
  "run_id": "example-run-id",
  "run_number": "10",
  "run_attempt": "2",
  "actor": "octocat",
  "workflow": "example-workflow",
  "head_ref": "",
  "base_ref": "",
  "event_name": "workflow_dispatch",
  "ref_type": "branch",
  "job_workflow_ref": "octo-org/octo-
automation/.github/workflows/oidc.yml@refs/heads/main",
  "iss": "https://token.actions.githubusercontent.com",
  "nbf": 1632492967,
  "exp": 1632493867,
  "iat": 1632493567
}
```


If your reusable workflow performs deployment steps, then it will typically need access to a specific cloud role, and you might want to allow any repository in your organization to call that reusable workflow. To permit this, you'll create the trust condition that allows any repository and any caller workflow, and then filter on the organization and the called workflow. See the next section for some examples.

Examples

Filtering for reusable workflows within a specific repository

You can configure a custom claim that filters for any reusable workflow in a specific repository. In this example, the workflow run must have originated from a job defined in a reusable workflow in the `octo-org/octo-automation` repository, and in any repository that is owned by the `octo-org` organization.

- **Subject**:
  - Syntax: `repo:ORG_NAME/*`

- Example: `repo:octo-org/*`

- **Custom claim**:
  - Syntax: `job_workflow_ref:ORG_NAME/REPO_NAME`
  - Example: `job_workflow_ref:octo-org/octo-automation@*`

Filtering for a specific reusable workflow at a specific ref

You can configure a custom claim that filters for a specific reusable workflow. In this example, the workflow run must have originated from a job defined in the reusable workflow `octo-org/octo-automation/.github/workflows/deployment.yml`, and in any repository that is owned by the `octo-org` organization.

- **Subject**:
  - Syntax: `repo:ORG_NAME/*`
  - Example: `repo:octo-org/*`

- **Custom claim**:
  - Syntax: `job_workflow_ref:ORG_NAME/REPO_NAME/.github/workflows/WORKFLOW_FILE@ref`
  - Example: `job_workflow_ref:octo-org/octo-automation/.github/workflows/deployment.yml@10040c56a8c0253d69db7c1f26a0d227275512e2`