

# when

Use `when` to configure the conditions for when jobs run. If not defined in a job, the default value is `when: on_success`.

**Keyword type:** Job keyword. You can use it as part of a job. `when: always` and `when: never` can also be used in `workflow:rules`.

## Possible inputs:

- `on_success` (default): Run the job only when all jobs in earlier stages succeed or have `allow_failure: true`.
- `manual`: Run the job only when [triggered manually](#).
- `always`: Run the job regardless of the status of jobs in earlier stages. Can also be used in `workflow:rules`.
- `on_failure`: Run the job only when at least one job in an earlier stage fails.
- `delayed`: [Delay the execution of a job](#) for a specified duration.

- `never`: Don't run the job. Can only be used in a `rules` section or `workflow: rules`.

### Example of `when`:

```
stages:
```

- `build`
- `cleanup_build`
- `test`
- `deploy`
- `cleanup`

```
build_job:
```

```
stage: build
```

```
script:
```

```
- make build
```

```
cleanup_build_job:
```

```
stage: cleanup_build
```

```
script:
```

```
- cleanup build when failed
```

```
when: on_failure
```

```
test_job:
```

```
  stage: test
```

```
  script:
```

```
    - make test
```

```
deploy_job:
```

```
  stage: deploy
```

```
  script:
```

```
  - make deploy

when: manual

cleanup_job:

  stage: cleanup

  script:

    - cleanup after jobs

when: always
```

In this example, the script:

1. Executes `cleanup_build_job` only when `build_job` fails.
2. Always executes `cleanup_job` as the last step in pipeline regardless of success or failure.
3. Executes `deploy_job` when you run it manually in the GitLab UI.

### **Additional details:**

- In [GitLab 13.5 and later](#), you can use `when:manual` in the same job as `trigger`. In GitLab 13.4 and earlier, using them together causes the error `jobs:#{job-name} when should be on_success, on_failure or always`.
- The default behavior of `allow_failure` changes to `true` with `when: manual`. However, if you use `when: manual` with `rules`, `allow_failure` defaults to `false`.

### **Related topics:**

- `when` can be used with `rules` for more dynamic job control.
- `when` can be used with `workflow` to control when a pipeline can start.

