stages

Use stages to define stages that contain groups of jobs. Use stage in a job to configure the job to run in a specific stage.

If stages is not defined in the .gitlab-ci.yml file, the default pipeline stages are:

- .pre
- build
- test
- deploy
- .post

The order of the items in stages defines the execution order for jobs:

- Jobs in the same stage run in parallel.
- Jobs in the next stage run after the jobs from the previous stage complete successfully.

If a pipeline contains only jobs in the .pre or .post stages, it does not run. There must be at least one other job in a different stage. .pre and .post stages can be used in required pipeline configuration to define compliance jobs that must run before or after project pipeline jobs.

Keyword type: Global keyword.

Example of stages:

```
stages:
  - build
  - test
  - deploy
```

In this example:

1. All jobs in build execute in parallel.
2. If all jobs in build succeed, the test jobs execute in parallel.
3. If all jobs in test succeed, the deploy jobs execute in parallel.
4. If all jobs in deploy succeed, the pipeline is marked as passed.

If any job fails, the pipeline is marked as failed and jobs in later stages do not start. Jobs in the current stage are not stopped and continue to run.

Additional details:

- If a job does not specify a stage, the job is assigned the test stage.
- If a stage is defined but no jobs use it, the stage is not visible in the pipeline, which can help compliance pipeline configurations:
  - Stages can be defined in the compliance configuration but remain hidden if not used.
  - The defined stages become visible when developers use them in job definitions.