



With a working project, we can finally get into implementing a CI/CD pipeline in GitLab. We'll use it to automatically run `dotnet build` and `dotnet test`. The pipeline configuration is defined in YAML, within the project itself. So, we have to create a new file in the root of the repo and commit it to GitLab.

```
PS C:\Tools\mycalculator> code .gitlab-ci.yml
```

```
image: mcr.microsoft.com/dotnet/sdk:6.0

variables:
  project: "Calculator"

stages:
  - build
  - test

before_script:
  - "dotnet restore"

build:
  stage: build
  variables:
    build_path: "$project"
  script:
    - "cd $build_path"
    - "dotnet build"

test:
  stage: test
  variables:
    test_path: "$project.Tests"
  script:
    - "cd $test_path"
    - "dotnet test"
```

The GitLab documentation includes a [keyword reference](#) for `.gitlab-ci.yml`.

Image defines the Docker container that we want the Runner to use. Because this is a .NET project, Microsoft's image makes logical sense as it will already contain the .NET SDK.

Stages allows us to break the pipeline down into smaller parts, which helps identify where any failures occur. E.g. did the pipeline failed because of a build failure, or because a test failure.

Within each stage, we have **script** options. These define the actual commands that need to be run. These are pretty much identical to what we ran manually in the last few lessons. However, you can see that it supports the use of **variables**, which makes re-using these configurations a little easier.

Push this file to the repo and the pipeline will trigger. The build stage will come first.

passed Job build triggered 1 minute ago by Rasta Mouse

```
Search job log
```

```

1 Running with gitlab-runner 15.5.1 (7178588d)
2   on gitlab yv1_4fbh
3   > Preparing the "docker" executor
4   > Preparing environment
5   > Getting source from Git repository
6   > Fetching changes with git depth set to 20...
7   > Initialized empty Git repository in /builds/rasta/mycalculator/.git/
8   > Created fresh repository.
9   > Checking out e45a9b9b as main...
10  > Skipping Git submodules setup
11  > Executing "step_script" stage of the job script
12  > Using docker image sha256:0ff04a23cb9e6263452b2cec8d7b23515163da2af5dd81f73263912baa71051e for mcr.microsoft.com/dotnet/sdk:6.0 with digest
13  > mcr.microsoft.com/dotnet/sdk@sha256:807eba67b95246174c63daa7f1d5b5990ce555ecbff73437bbe6c31f28660acb ...
14  > $ dotnet restore
15  > Determining projects to restore...
16  > Restored /builds/rasta/mycalculator/Calculator/Calculator.csproj (in 103 ms).
17  > Restored /builds/rasta/mycalculator/Calculator.Tests/Calculator.Tests.csproj (in 45.78 sec).
18  > $ cd $build_path
19  > $ dotnet build
20  > MSBuild version 17.3.2+561848881 for .NET
21  > Determining projects to restore...
22  > All projects are up-to-date for restore.
23  > Calculator -> /builds/rasta/mycalculator/Calculator/bin/Debug/net6.0/Calculator.dll
24  > Build succeeded.
25  > 0 Warning(s)
26  > 0 Error(s)
27  > Time Elapsed 00:00:02.24
28  > Job succeeded

```

Followed by the test stage.

passed Job test triggered just now by Rasta Mouse

```
Search job log
```

```

1 Running with gitlab-runner 15.5.1 (7178588d)
2   on gitlab yv1_4fbh
3   > Preparing the "docker" executor
4   > Preparing environment
5   > Running on runner-yv14fbh-project-3-concurrent-0 via gitlab...
6   > Getting source from Git repository
7   > Executing "step_script" stage of the job script
8   > Using docker image sha256:0ff04a23cb9e6263452b2cec8d7b23515163da2af5dd81f73263912baa71051e for mcr.microsoft.com/dotnet/sdk:6.0 with digest
9   > mcr.microsoft.com/dotnet/sdk@sha256:807eba67b95246174c63daa7f1d5b5990ce555ecbff73437bbe6c31f28660acb ...
10  > $ dotnet restore
11  > Determining projects to restore...
12  > Restored /builds/rasta/mycalculator/Calculator/Calculator.csproj (in 87 ms).
13  > Restored /builds/rasta/mycalculator/Calculator.Tests/Calculator.Tests.csproj (in 35.53 sec).
14  > $ cd $test_path
15  > $ dotnet test
16  > Determining projects to restore...
17  > All projects are up-to-date for restore.
18  > Calculator -> /builds/rasta/mycalculator/Calculator/bin/Debug/net6.0/Calculator.dll
19  > Calculator.Tests -> /builds/rasta/mycalculator/Calculator.Tests/bin/Debug/net6.0/Calculator.Tests.dll
20  > Test run for /builds/rasta/mycalculator/Calculator.Tests/bin/Debug/net6.0/Calculator.Tests.dll (.NETCoreApp,Version=v6.0)
21  > Microsoft (R) Test Execution Command Line Tool Version 17.3.1 (x64)
22  > Copyright (c) Microsoft Corporation. All rights reserved.
23  > Starting test execution, please wait...
24  > A total of 1 test files matched the specified pattern.
25  > Passed! - Failed: 0, Passed: 4, Skipped: 0, Total: 4, Duration: 4 ms - /builds/rasta/mycalculator/Calculator.Tests/bin/Debug/net6.0/Calculator.Tests.dll (net6.0)
26  > Job succeeded

```

Hopefully everything will pass, and you'll see a lot of satisfying green ticks.