Step 1 - Create a repository

1. Log in to GitHub and begin the process to create a new repository.
2. Enter a name for your repository (for example, hello-world).
3. Select the option to initialize the repository with a README file.
4. Finally, click Create repository.
5. There is no need to add any source code for now.

Login to Circle CI https://app.circleci.com/ Using GitHub Login, Once logged in navigate to Projects.

Step 2 - Set up CircleCI

1. Navigate to the CircleCI Projects page. If you created your new repository under an organization, you will need to select the organization name.
2. You will be taken to the Projects dashboard. On the dashboard, select the project you want to set up (hello-world).
3. Select the option to commit a starter CI pipeline to a new branch, and click Set Up Project. This will create a file .circleci/config.yml at the root of your repository on a new branch called circleci-project-setup.
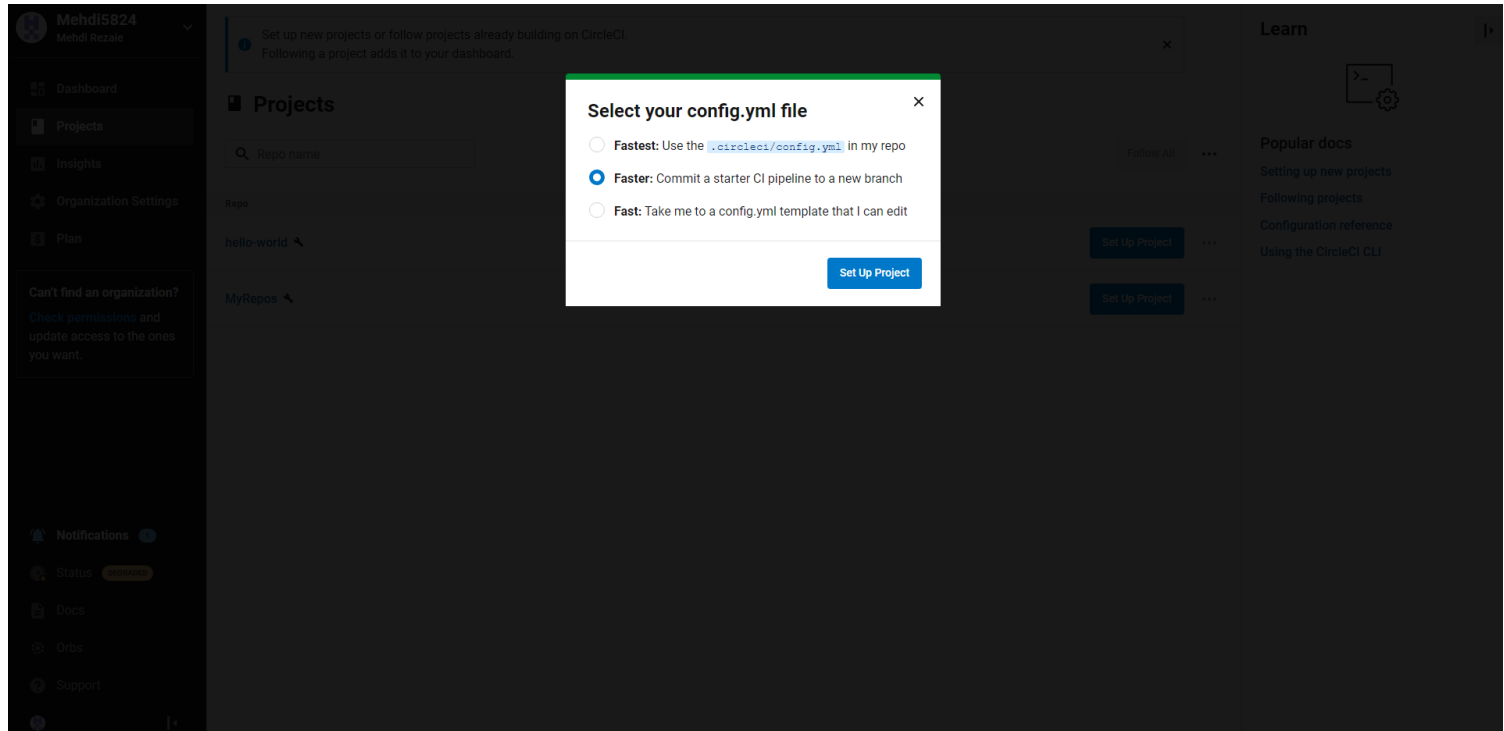
Step 3 - Your first pipeline

On your project's pipeline page, click the green Success button, which brings you to the workflow that ran (say-hello-workflow).

Within this workflow, the pipeline ran one job, called say-hello. Click say-hello to see the steps in this job:

a. Spin up environment
b. Preparing environment variables
c. Checkout code
d. Say hello

Now select the "say-hello-workflow" to the right of Success status column

hello-world · Add team members

Edit Config | Trigger Pipeline | Project Settings

Filters

Everyone's Pipelines ▾ | hello-world ▾ | All Branches ▾ | ▾

Auto-expand ⬤

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 1	✓ Success	say-hello-workflow	circleci-project-setup 6b308f0	3m ago	5s	⟳ ⟲ ⊗ ⋯

Select "say-hello" Job with a green tick

say-hello-workflow ✓ Success

Insights | ⟳ Rerun ▾ | ⋯

Duration / Finished	Branch	Commit	Author
⏱ 5s / 8m ago	circleci-project-setup	⊸ 6b308f0	Mehdi5824

✓ say-hello   3s

Did you know? CircleCI teams that commit 4x as often fix failed builds 2x faster.

20 pipelines/day   70 min to recovery
5 pipelines/day   142 min

Learn more ✕

Select Branch and option circleci-project-setup

Step 4 - Break your build

In this section, you will edit the .circleci/config.yml file and see what happens if a build does not complete successfully.

It is possible to edit files directly on GitHub.

Search or jump to...

Pull requests Issues Marketplace Explore

Mehdi5824 / **hello-world** Public

Pin   Unwatch 1   Fork 0   Star 0

<> Code   Issues   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

circleci-project-setup had recent pushes 14 minutes ago

Compare & pull request

circleci-proje...   2 branches   0 tags

Go to file   Add file   Code

This branch is 1 commit ahead of main.

Contribute

Mehdi5824 Add .circleci/config.yml   6b308f0 14 minutes ago   2 commits

.circleci   Add .circleci/config.yml   14 minutes ago

README.md   Initial commit   2 hours ago

README.md

hello-world

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

© 2022 GitHub, Inc.   Terms   Privacy   Security   Status   Docs   Contact GitHub   Pricing   API   Training   Blog   About

---

Search or jump to...

Pull requests Issues Marketplace Explore

Mehdi5824 / **hello-world** Public

Pin   Unwatch 1

<> Code   Issues   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

circleci-proje...   hello-world / .circleci /

Go to file   Add file   ...

This branch is 1 commit ahead of main.

Contribute

Mehdi5824 Add .circleci/config.yml   6b308f0 15 minutes ago   History

..

config.yml   Add .circleci/config.yml   15 minutes ago

© 2022 GitHub, Inc.   Terms   Privacy   Security   Status   Docs   Contact GitHub   Pricing   API   Training   Blog   About

---

Search or jump to...

Pull requests Issues Marketplace Explore

Mehdi5824 / **hello-world** Public

Pin   Unwatch 1   Fork 0   Star 0

<> Code   Issues   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

circleci-proje...   hello-world / .circleci / config.yml

Go to file   ...

Mehdi5824 Add .circleci/config.yml   Latest commit 6b308f0 16 minutes ago   History

1 contributor

26 lines (24 sloc) 944 Bytes

Raw Blame

```
1   # Use the latest 2.1 version of CircleCI pipeline process engine.
2   # See: https://circleci.com/docs/2.0/configuration-reference
3   version: 2.1
4
5   # Define a job to be invoked later in a workflow.
6   # See: https://circleci.com/docs/2.0/configuration-reference/#jobs
7   jobs:
8     say-hello:
9       # Specify the execution environment. You can specify an image from Dockerhub or use one of our Convenience Images from CircleCI's Developer Hub.
10      # See: https://circleci.com/docs/2.0/configuration-reference/#docker-machine-macos-windows-executor
11      docker:
12        - image: cimg/base:stable
13      # Add steps to the job
14      # See: https://circleci.com/docs/2.0/configuration-reference/#steps
15      steps:
16        - checkout
17        - run:
18            name: "Say hello"
19            command: "echo Hello, World!"
20
21  # Invoke jobs via workflows
22  # See: https://circleci.com/docs/2.0/configuration-reference/#workflows
23  workflows:
24    say-hello-workflow:
25      jobs:
26        - say-hello
```

Let's use the Node orb. Replace the existing config by pasting the following code:

```
1    version: 2.1
2    orbs:
3      node: circleci/node@4.7.0
4    jobs:
5      build:
6        executor:
7          name: node/default
8          tag: '10.4'
9        steps:
10         - checkout
11         - node/with-cache:
12             steps:
13               - run: npm install
14         - run: npm run test
```

The GitHub file editor should look like this
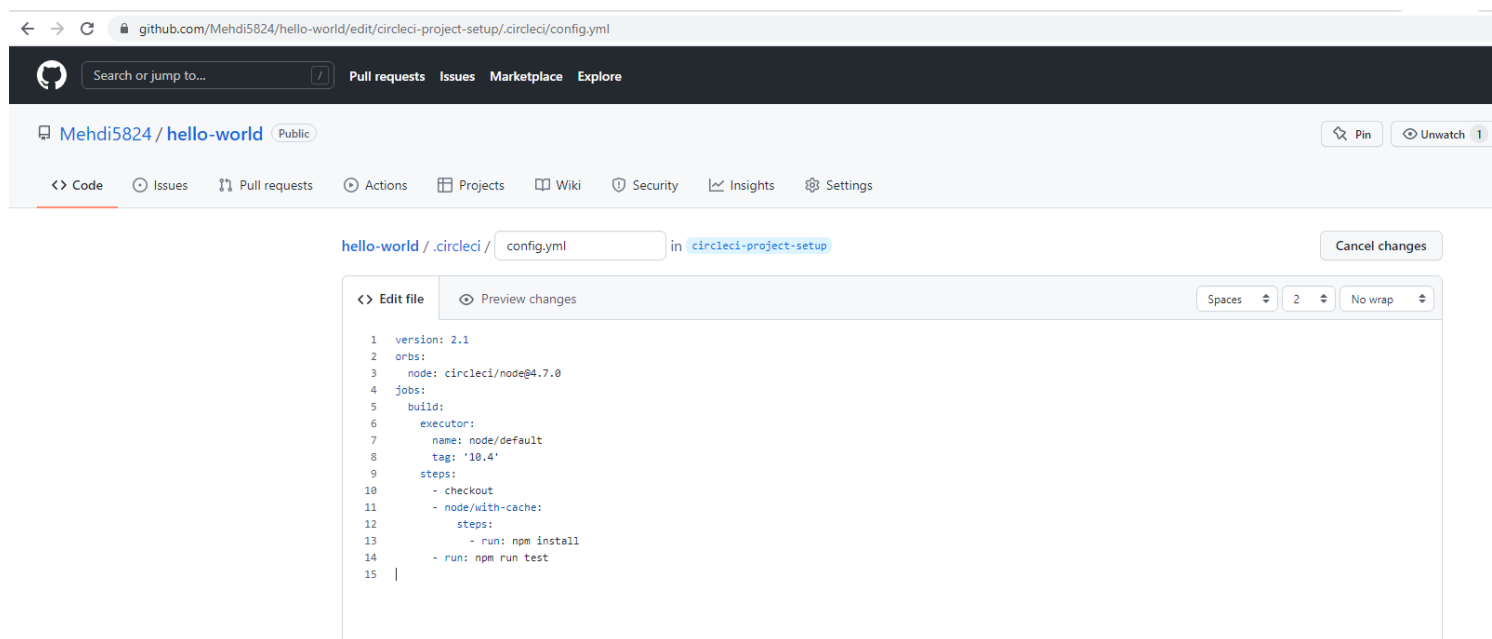
Scroll down and Commit your changes on GitHub

```
10      - checkout
11      - node/with-cache:
12          steps:
13            - run: npm install
14      - run: npm run test
15
```

Commit changes

Update config.yml

we have updated to config.yml file

○ ⊶ Commit directly to the `circleci-project-setup` branch.
○ ⇅ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Commit changes   **Cancel**

After committing your changes, then return to the Projects page in CircleCI. You should see a new pipeline running... and it will fail! What's going on? The Node orb runs some common Node tasks. Because you are working with an empty repository, running npm run test, a Node script, causes the configuration to fail. To fix this, you need to set up a Node project in your repository.

Mehdi5824
Mehdi Rezaie

- Dashboard
- Projects
- Insights
- Organization Settings
- Plan UPGRADE

Can't find an organization?
Check permissions and update access to the ones you want.

Dashboard
▣ All Pipelines

▣ All Pipelines

| ⧉ Everyone's Pipelines ▾ | ▣ All Projects ▾ | ⑂ All Branches ▾ | ▽ ▾ | | Auto-expand ◯ |

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 2		⊗ Error calling workflow: 'workflow'Error calling job: 'build'Cannot find a definition for command named node/with-cache				
	🔴 Failed	Build Error	�H circleci-project-setup ca5ab74 Update config.yml	3m ago	6s	↺ ↺ₓ ⊗ ⋯
hello-world 1	✅ Success	say-hello-workflow	�H circleci-project-setup 6b308f0	22m ago	⊪ 5s	↺ ↺ₓ ⊗ ⋯

No more pipelines to load

🔔 Notifications ①
🔂 Status DEGRADED
📄 Docs
⚙ Orbs
❓ Support

Step 5 – Use Workflows

You do not have to use orbs to use CircleCI. The following example details how to create a custom configuration that also uses the workflow feature of CircleCI.

1) Take a moment and read the comments in the code block below. Then, to see workflows in action, edit your **.circleci/config.yml file** and copy and paste the following text into it.

```
1   version: 2
2   jobs: # we now have TWO jobs, so that a workflow can coordinate them!
3     one: # This is our first job.
4       docker: # it uses the docker executor
5         - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
6           auth:
7             username: mydockerhub-user
8             password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
9       # Steps are a list of commands to run inside the docker container above.
10      steps:
11        - checkout # this pulls code down from GitHub
12        - run: echo "A first hello" # This prints "A first hello" to stdout.
13        - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
14    two: # This is our second job.
15      docker: # it runs inside a docker image, the same as above.
16        - image: cimg/ruby:3.0.2
17          auth:
18            username: mydockerhub-user
19            password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
20      steps:
21        - checkout
22        - run: echo "A more familiar hi" # We run a similar echo command to above.
23        - run: sleep 15 # and then sleep for 15 seconds.
24  # Under the workflows: map, we can coordinate our two jobs, defined above.
25  workflows:
26    version: 2
27    one_and_two: # this is the name of our workflow
28      jobs: # and here we list the jobs we are going to run.
29        - one
30        - two
```

You don't need to write the comments which are the text after #

2) Commit these changes to your repository and navigate back to the CircleCI Pipelines page. You should see your pipeline running.

⇅ Pull requests   ⊙ Actions   ⊞ Projects   ▭ Wiki   ⊘ Security   ⎘ Insights   ⚙ Settings

hello-world / .circleci / `config.yml`   in `circleci-project-setup`   Cancel changes

<> Edit file   ⊙ Preview changes   Spaces ⇕   2 ⇕   No wrap ⇕

```
 1   version: 2
 2   jobs: # we now have TWO jobs, so that a workflow can coordinate them!
 3     one: # This is our first job.
 4       docker: # it uses the docker executor
 5         - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
 6           auth:
 7             username: mydockerhub-user
 8             password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
 9       # Steps are a list of commands to run inside the docker container above.
10       steps:
11         - checkout # this pulls code down from GitHub
12         - run: echo "A first hello" # This prints "A first hello" to stdout.
13         - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
14     two: # This is our second job.
15       docker: # it runs inside a docker image, the same as above.
16         - image: cimg/ruby:3.0.2
17           auth:
18             username: mydockerhub-user
19             password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
20       steps:
21         - checkout
22         - run: echo "A more familiar hi" # We run a similar echo command to above.
23         - run: sleep 15 # and then sleep for 15 seconds.
24   # Under the workflows: map, we can coordinate our two jobs, defined above.
25   workflows:
26     version: 2
27     one_and_two: # this is the name of our workflow
28       jobs: # and here we list the jobs we are going to run.
29         - one
30         - two
31
```

3) Click on the running pipeline to view the workflow you have created. You should see that two jobs ran (or are currently running!) concurrently.

Step 5 – Add some changes to use workspaces

Each workflow has an associated workspace which can be used to transfer files to downstream jobs as the workflow progresses. You can use workspaces to pass along data that is unique to this run and which is needed for downstream jobs. Try updating config.yml to the following:

```
1   version: 2
2   jobs:
3     one:
4       docker:
5         - image: cimg/ruby:3.0.2
6           auth:
7             username: mydockerhub-user
8             password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
9       steps:
10        - checkout
11        - run: echo "A first hello"
12        - run: mkdir -p my_workspace
13        - run: echo "Trying out workspaces" > my_workspace/echo-output
14        - persist_to_workspace:
15            # Must be an absolute path, or relative path from working_directory
16            root: my_workspace
17            # Must be relative path from root
18            paths:
19              - echo-output
20    two:
21      docker:
22        - image: cimg/ruby:3.0.2
23          auth:
24            username: mydockerhub-user
25            password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
26      steps:
27        - checkout
28        - run: echo "A more familiar hi"
29        - attach_workspace:
30            # Must be absolute path or relative path from working_directory
31            at: my_workspace
```

```
32
33        - run: |
34            if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
35              echo "It worked!";
36            else
37              echo "Nope!"; exit 1
38            fi
39  workflows:
40    version: 2
41    one_and_two:
42      jobs:
43        - one
44        - two:
45            requires:
46              - one
```

Updated config.yml in GitHub file editor should be updated like this

hello-world / .circleci / config.yml   in circleci-project-setup   Cancel changes

<> Edit file   ⊙ Preview changes   Spaces ▼   2 ▼   No wrap ▼

```
 1  version: 2
 2  jobs:
 3    one:
 4      docker:
 5        - image: cimg/ruby:3.0.2
 6          auth:
 7            username: mydockerhub-user
 8            password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
 9      steps:
10        - checkout
11        - run: echo "A first hello"
12        - run: mkdir -p my_workspace
13        - run: echo "Trying out workspaces" > my_workspace/echo-output
14        - persist_to_workspace:
15            # Must be an absolute path, or relative path from working_directory
16            root: my_workspace
17            # Must be relative path from root
18            paths:
19              - echo-output
20    two:
21      docker:
22        - image: cimg/ruby:3.0.2
23          auth:
24            username: mydockerhub-user
25            password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
26      steps:
27        - checkout
28        - run: echo "A more familiar hi"
29        - attach_workspace:
30            # Must be absolute path or relative path from working_directory
31            at: my_workspace
32
33        - run: |
34            if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
35              echo "It worked!";
```

H | Commit changes

H

Commit changes

Update config.yml

3rd Update

○ ⊶ Commit directly to the circleci-project-setup branch.
○ ⊶ Create a **new branch** for this commit and start a pull request. Learn more about pull requests.

Commit changes   Cancel

Finally your workflow with the jobs running should look like this