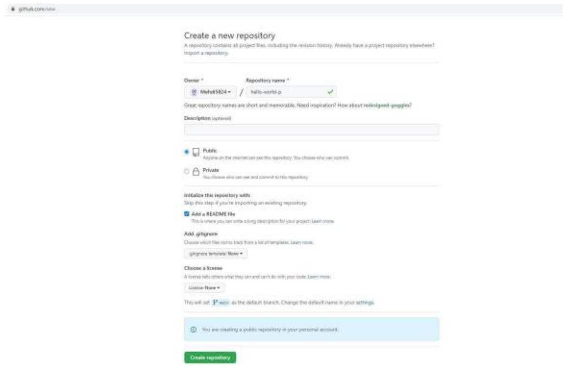
Practical No.6

Aim: Working with Circle CI for continuous integration.

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.
There is no need to add any source code for now.


Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Mehdi5824 / Repository name * hello-world-p

Great repository names are short and memorable. Need inspiration? How about redesigned-goggles?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

[✓ Add a README file] This is where you can write a long description for your project. Learn more.

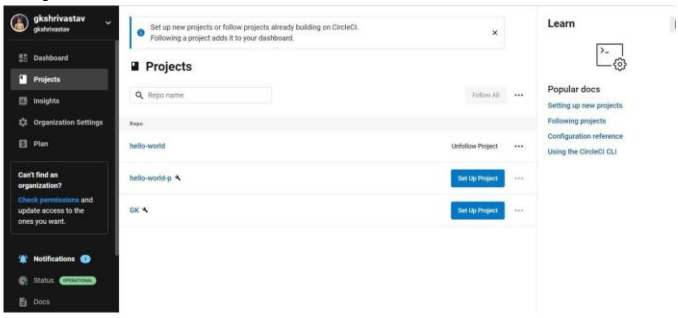
Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

This will set main as the default branch. Change the default name in your settings.

You are creating a public repository in your personal account.

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


gkshrivastav
gkshrivastav

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

Set up new projects or follow projects already building on CircleCI. Following a project adds it to your dashboard.

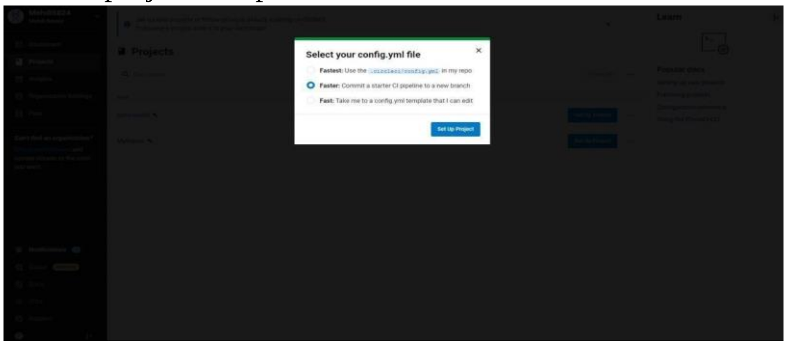
Projects

Repo
hello-world — Unfollow Project
hello-world-p — Set Up Project
GK — Set Up Project

Learn

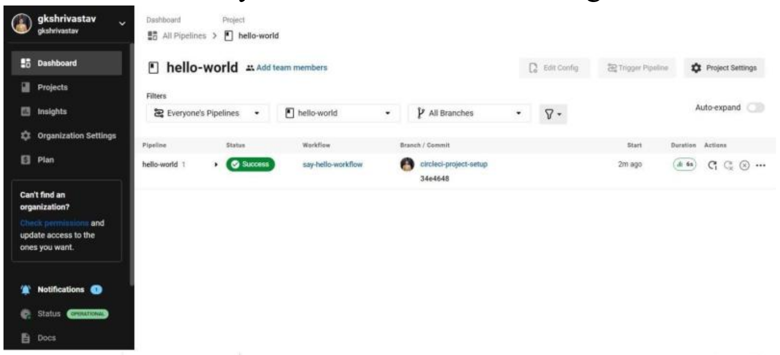
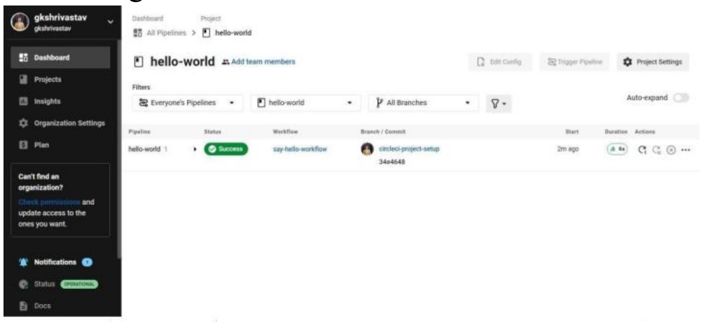
Popular docs
Setting up new projects
Following projects
Configuration reference
Using the CircleCI CLI

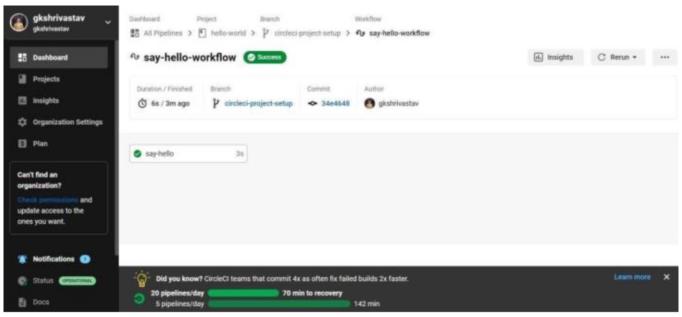
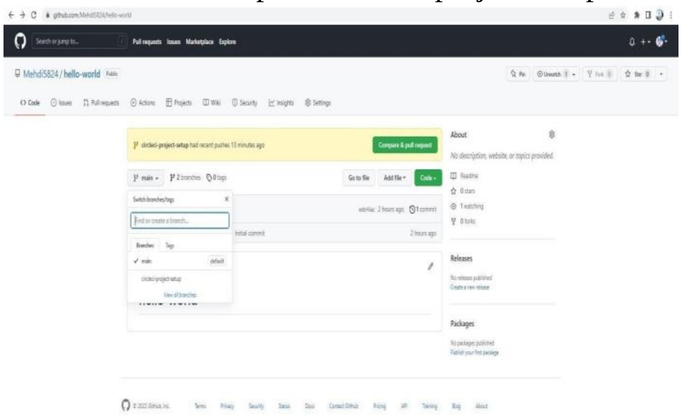
Step 2 - Set up CircleCI
1. Navigate to the CircleCI Projects page. If you created your new repository under an organization, you will need to select the organization name.
2. You will be taken to the Projects dashboard. On the dashboard,select the project you want to set up (hello-world).
Select the 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

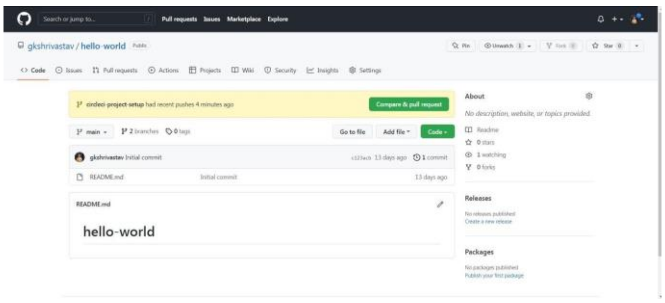

Select your config.yml file

Fastest: Use the .circleci/config.yml in my repo
[✓ Faster: Commit a starter CI pipeline to a new branch]
Fast: Take me to a config.yml template that I can edit

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

gkshrivastav
Dashboard
Projects
Insights
Organization Settings
Plan
Can't find an organization?
Check permissions and update access to the ones you want.
Notifications
Status OPERATIONAL
Docs
All Pipelines > hello-world
hello-world
Add team members
Edit Config
Trigger Pipeline
Project Settings
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 34e4648 2m ago 6s
Select "say-hello" Job with a green tick



All Pipelines > hello-world > circleci-project-setup > say-hello-workflow
say-hello-workflow Success
Insights
Rerun
Duration / Finished 6s / 3m ago
Branch circleci-project-setup
Commit 34e4648
Author gkshrivastav
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

github.com/Mehdi5824/hello-world
Pull requests Issues Marketplace Explore
Mehdi5824 / hello-world Public
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
circleci-project-setup had recent pushes 13 minutes ago
Compare & pull request
main 2 branches 0 tags
Go to file Add file Code
Switch branches/tags
Find or create a branch...
Branches Tags
main default
circleci-project-setup
View all branches
About
No description, website, or topics provided.
Readme
0 stars
1 watching
0 forks
Releases
Packages
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
gkshrivastav / hello-world Public
Pin Unwatch 1 Fork 0 Star 0
Code Issues Pull requests Actions Projects Wiki Security Insights Settings
Compare & pull request
main 2 branches 0 tags
Go to file Add file Code

circleci-project-setup had recent pushes 4 minutes ago

gkshrivastav Initial commit c123ach 13 days ago 1 commit

README.md Initial commit 13 days ago

README.md

hello-world

About

No description, website, or topics provided.

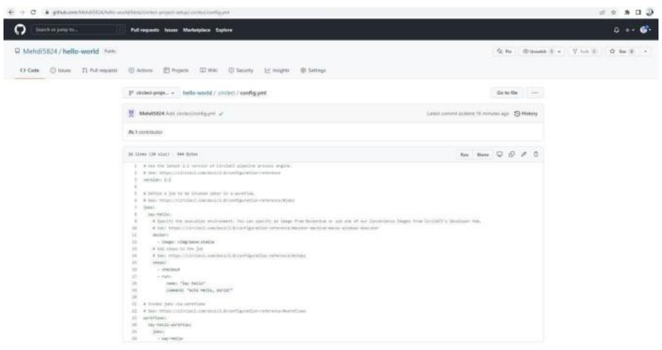
Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package


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...
Go to file
History
Raw Blame

hello-world / .circleci / config.yml

Mehdi5824 Add .circleci/config.yml ✓ Latest commit 4cda9fe 15 minutes ago

1 contributor

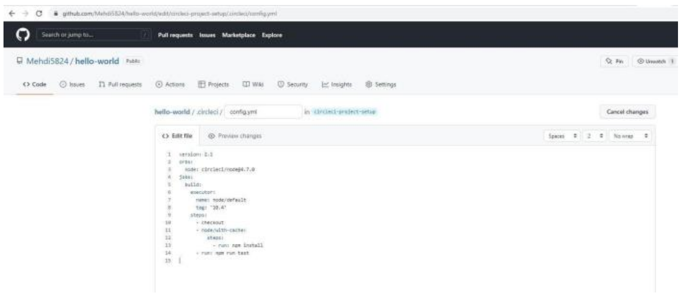
26 lines (24 sloc) 944 Bytes

```
# Use the latest 2.1 version of CircleCI pipeline process engine.
# See: https://circleci.com/docs/2.0/configuration-reference
version: 2.1

# Define a job to be invoked later in a workflow.
# See: https://circleci.com/docs/2.0/configuration-reference/#jobs
jobs:
  say-hello:
    # Specify the execution environment. You can specify an image from Dockerhub or use one of our Convenience Images from CircleCI's Developer Hub.
    # See: https://circleci.com/docs/2.0/configuration-reference/#docker-machine-macos-windows-executor
    docker:
      - image: cimg/base:stable
    # Add steps to the job
    # See: https://circleci.com/docs/2.0/configuration-reference/#steps
    steps:
      - checkout
      - run:
          name: "Say hello"
          command: "echo Hello, World!"

# Invoke jobs via workflows
# See: https://circleci.com/docs/2.0/configuration-reference/#workflows
workflows:
  say-hello-workflow:
    jobs:
      - say-hello
```

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

```
version: 2.1
orbs:
  node: circleci/node@4.7.0
jobs:
  build:
    executor:
      name: node/default
      tag: '10.4'
    steps:
      - checkout
      - node/with-cache:
          steps:
            - run: npm install
      - run: npm run test
```

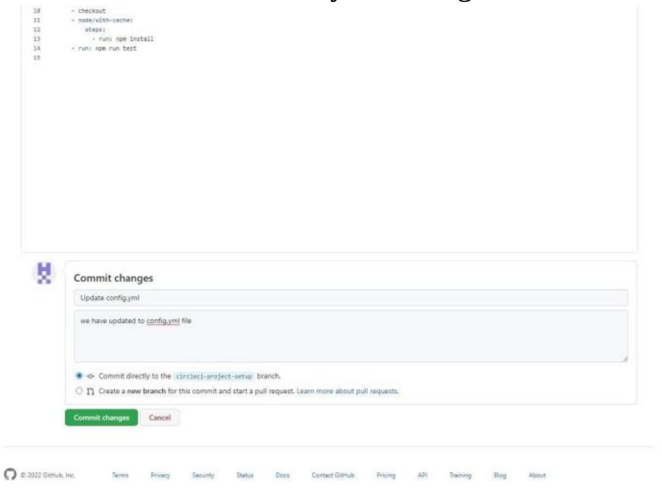
The GitHub file editor should look like this


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
Cancel changes
Edit file
Preview changes
Spaces 2 No wrap

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

```
version: 2.1
orbs:
  node: circleci/node@4.7.0
jobs:
  build:
    executor:
      name: node/default
      tag: '10.4'
    steps:
      - checkout
      - node/with-cache:
          steps:
            - run: npm install
      - run: npm run test

```

Scroll down and Commit your changes on GitHub


Commit changes
Cancel
Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

```
      - checkout
      - node/with-cache:
          steps:
            - run: npm install
      - run: npm run test

```

Commit changes

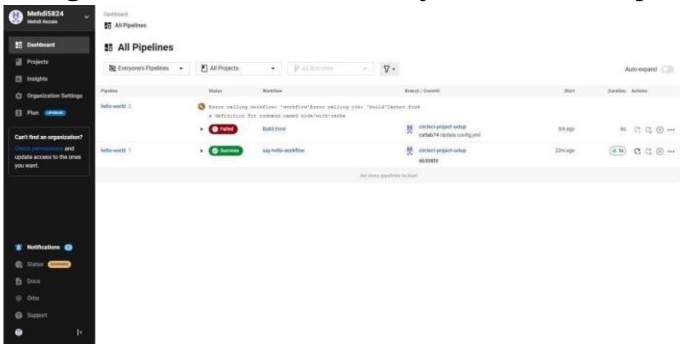
Update config.yml

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.

© 2022 GitHub, Inc.

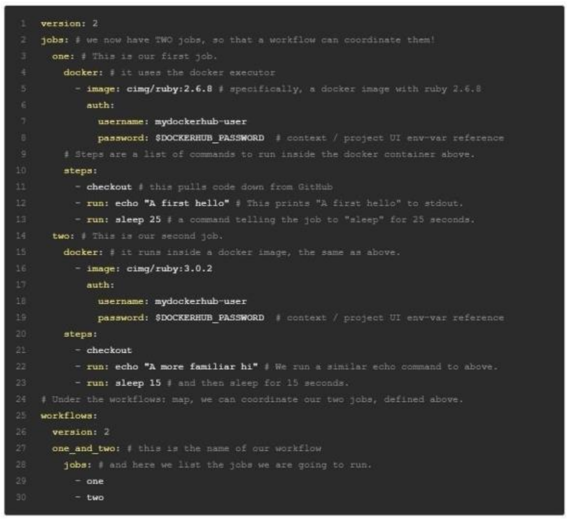
After committing your changes, then return to the Projects page in CircleCI. You should see a new pipeline running and I t will fail! What’s going on? The Node orb runs some common Node
tasks. Because you are work in empty repository, running npm run test, a Node script, causes the
configuration to fail. To fix this, you need to set up a Node project in your repository.



hello-world 2 — Error calling workflow: 'workflow'Error calling jobs: 'build'Cannot find a definition for command named node/with-cache
Failed — Build Error — circleci-project-setup ca5ab74 Update config.yml — 9m ago
hello-world 1 — Success — say-hello-workflow — circleci-project-setup 6630818 — 22m ago

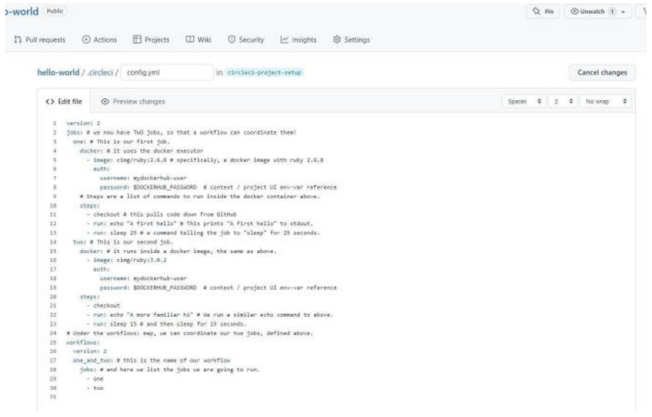
No more pipelines to load

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.
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.



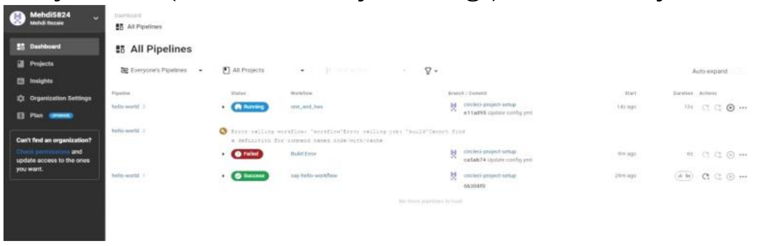
```yaml
version: 2
jobs: # we now have TWO jobs, so that a workflow can coordinate them!
  one: # This is our first job.
    docker: # it uses the docker executor
      - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    # Steps are a list of commands to run inside the docker container above.
    steps:
      - checkout # this pulls code down from GitHub
      - run: echo "A first hello" # This prints "A first hello" to stdout.
      - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
  two: # This is our second job.
    docker: # it runs inside a docker image, the same as above.
      - image: cimg/ruby:3.0.2
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A more familiar hi" # We run a similar echo command to above.
      - run: sleep 15 # and then sleep for 15 seconds.
# Under the workflows: map, we can coordinate our two jobs, defined above.
workflows:
  version: 2
  one_and_two: # this is the name of our workflow
    jobs: # and here we list the jobs we are going to run.
      - one
      - 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.



```yaml
version: 2
jobs: # we now have TWO jobs, so that a workflow can coordinate them!
  one: # This is our first job.
    docker: # it uses the docker executor
      - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    # Steps are a list of commands to run inside the docker container above.
    steps:
      - checkout # this pulls code down from GitHub
      - run: echo "A first hello" # This prints "A first hello" to stdout.
      - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
  two: # This is our second job.
    docker: # it runs inside a docker image, the same as above.
      - image: cimg/ruby:3.0.2
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A more familiar hi" # We run a similar echo command to above.
      - run: sleep 15 # and then sleep for 15 seconds.
# Under the workflows: map, we can coordinate our two jobs, defined above.
workflows:
  version: 2
  one_and_two: # this is the name of our workflow
    jobs: # and here we list the jobs we are going to run.
      - one
      - two
```

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



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

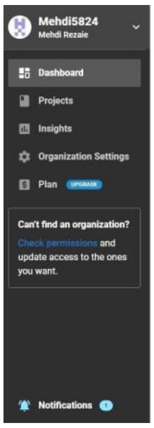
All Pipelines

Everyone's Pipelines | All Projects | Auto-expand

Pipeline	Status	Workflow	Branch / Commit	Start	Duration	Actions
hello-world 3	Running	one_and_two	circleci-project-setup e11ad95 update config.yml	14s ago	13s	
hello-world 2	Failed	Build Error	circleci-project-setup ca5ab74 update config.yml	9m ago	6s	
hello-world 1	Success	say-hello-workflow	circleci-project-setup 6630890	29m ago		

Error calling workflow: 'workflow'Error calling job: 'build'Cannot find a definition for command named node/with-cache

No more pipelines to load



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.

Notifications

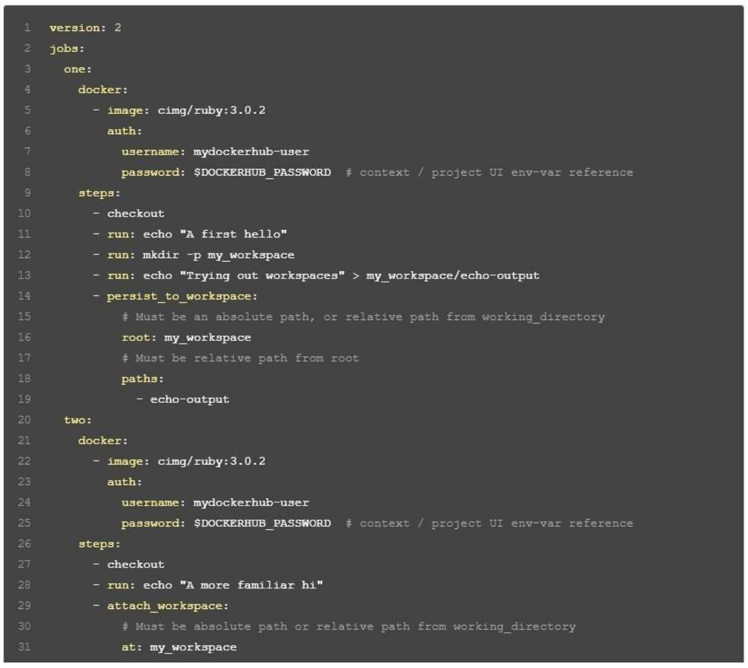
Dashboard > Project > Branch > Workflow
All Pipelines > hello-world > circleci-project-setup > one_and_two

one_and_two Success

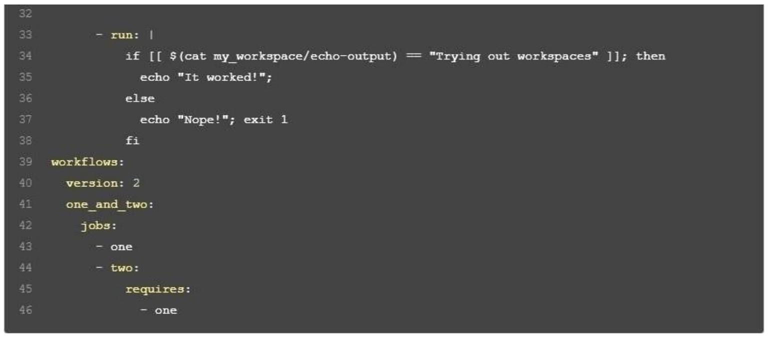
Duration / Finished	Branch	Commit	Author & Message
34s / 3s ago	circleci-project-setup	e11ad95	Update config.yml

two 19s
one 31s

Step 5 – Add some changes to use workspaces.
Each workflow has an associated workspace which can be used to transfer files to down stream.
jobs as the workflow progress. You can use workspaces to pass along data that is unique to this run and which needed for down jobs. updating config.yml to the following

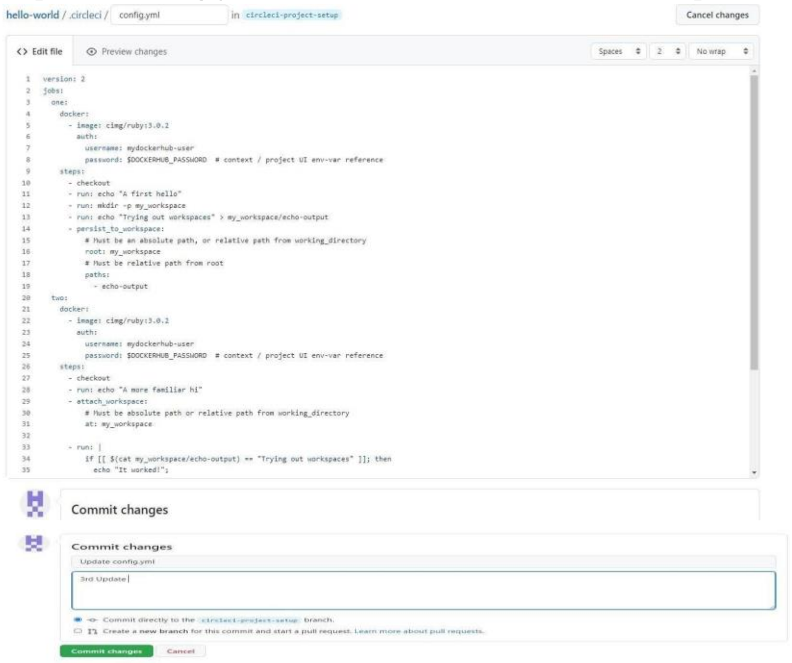


```yaml
version: 2
jobs:
  one:
    docker:
      - image: cimg/ruby:3.0.2
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A first hello"
      - run: mkdir -p my_workspace
      - run: echo "Trying out workspaces" > my_workspace/echo-output
      - persist_to_workspace:
          # Must be an absolute path, or relative path from working_directory
          root: my_workspace
          # Must be relative path from root
          paths:
            - echo-output
  two:
    docker:
      - image: cimg/ruby:3.0.2
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A more familiar hi"
      - attach_workspace:
          # Must be absolute path or relative path from working_directory
          at: my_workspace
```



```
      - run: |
          if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
            echo "It worked!";
          else
            echo "Nope!"; exit 1
          fi
workflows:
  version: 2
  one_and_two:
    jobs:
      - one
      - two:
          requires:
            - 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

```
version: 2
jobs:
  one:
    docker:
      - image: cimg/ruby:3.0.2
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A first hello"
      - run: mkdir -p my_workspace
      - run: echo "Trying out workspaces" > my_workspace/echo-output
      - persist_to_workspace:
          # Must be an absolute path, or relative path from working_directory
          root: my_workspace
          # Must be relative path from root
          paths:
            - echo-output
  two:
    docker:
      - image: cimg/ruby:3.0.2
        auth:
          username: mydockerhub-user
          password: $DOCKERHUB_PASSWORD  # context / project UI env-var reference
    steps:
      - checkout
      - run: echo "A more familiar hi"
      - attach_workspace:
          # Must be absolute path or relative path from working_directory
          at: my_workspace

      - run: |
          if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
            echo "It worked!";
```

Commit changes

Commit changes

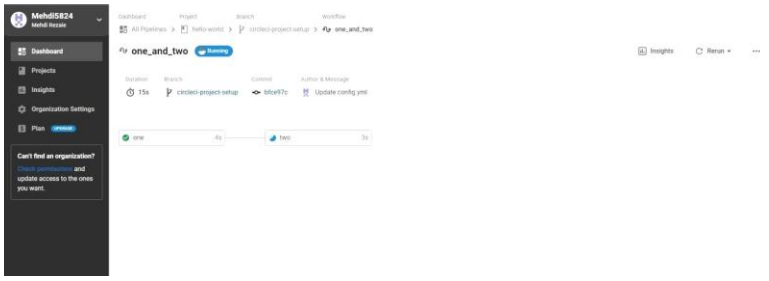
Update config.yml

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.

Finally your workflow with the jobs running should look like this.



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