

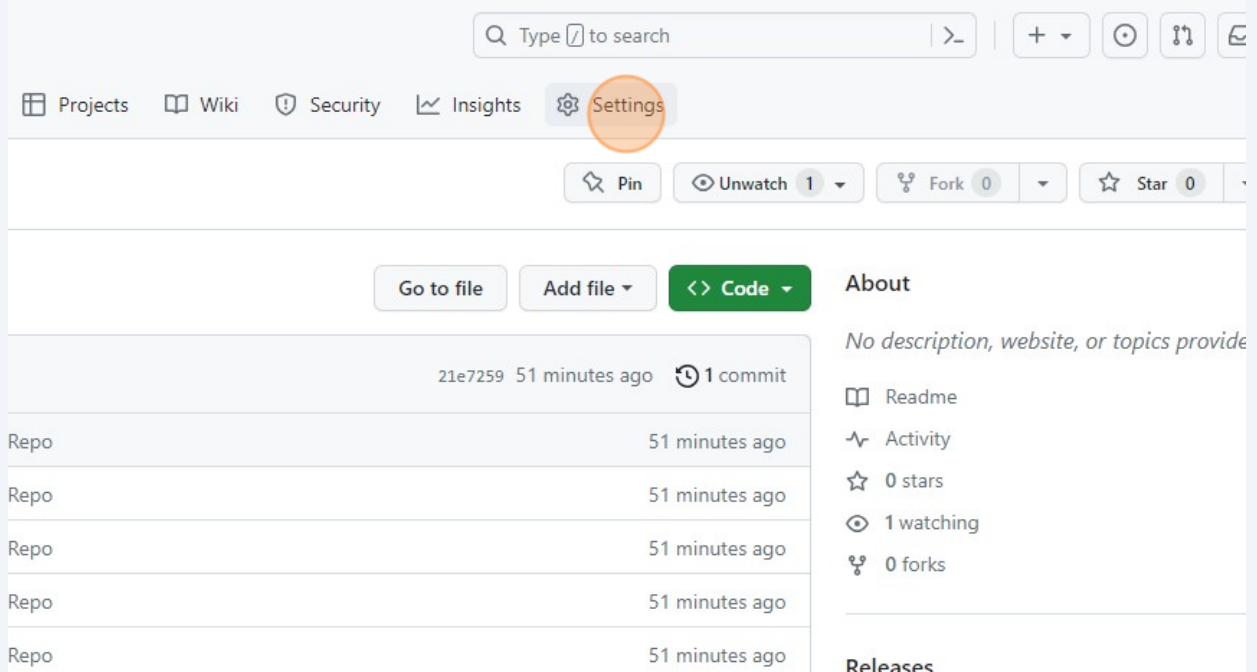
Deploy React Sample App to EC2 Instance using Github Actions

1

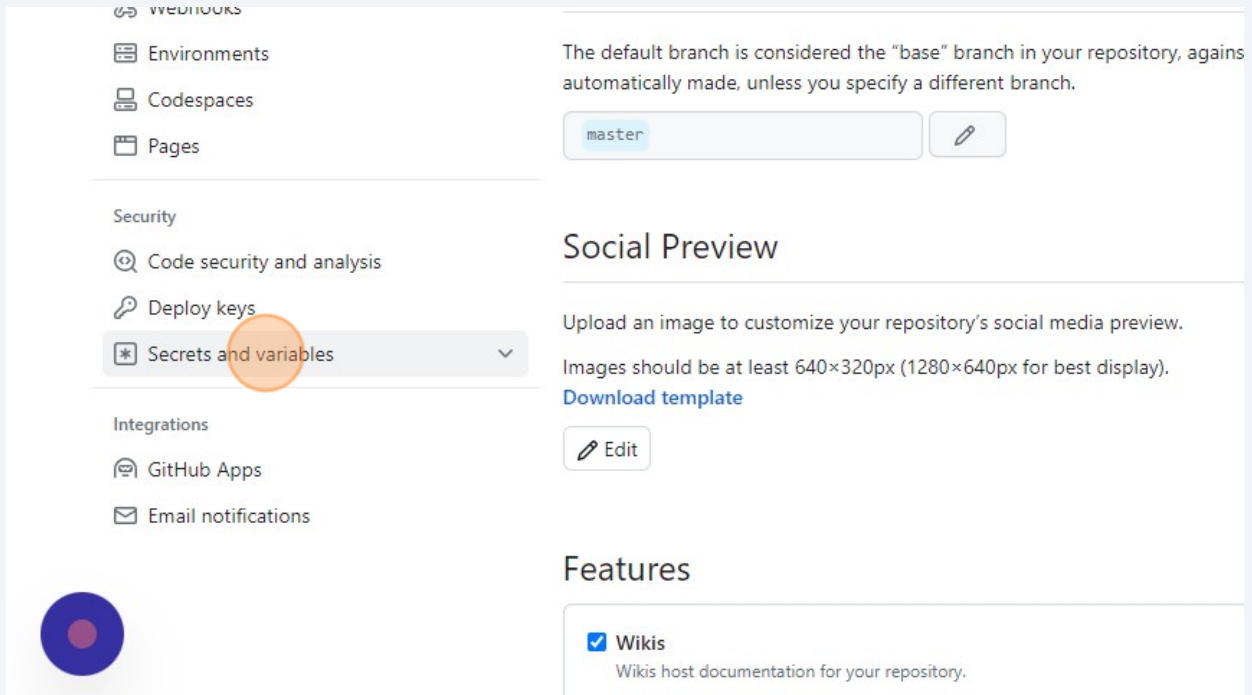
Navigate to github.com/thomasjv799/react-sample-app and clone this repository to your github.

2

Click "Settings"



3 Click "Secrets and variables"



The screenshot shows the GitHub repository settings page. On the left sidebar, under the 'Security' section, 'Secrets and variables' is highlighted with an orange circle. The main content area shows the 'master' branch selected in a dropdown menu. Below this, there are sections for 'Social Preview' and 'Features'. The 'Social Preview' section includes a description, a 'Download template' link, and an 'Edit' button. The 'Features' section shows the 'Wikis' feature is enabled.

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

The default branch is considered the "base" branch in your repository, against automatically made, unless you specify a different branch.

master

Social Preview

Upload an image to customize your repository's social media preview. Images should be at least 640×320px (1280×640px for best display). [Download template](#)

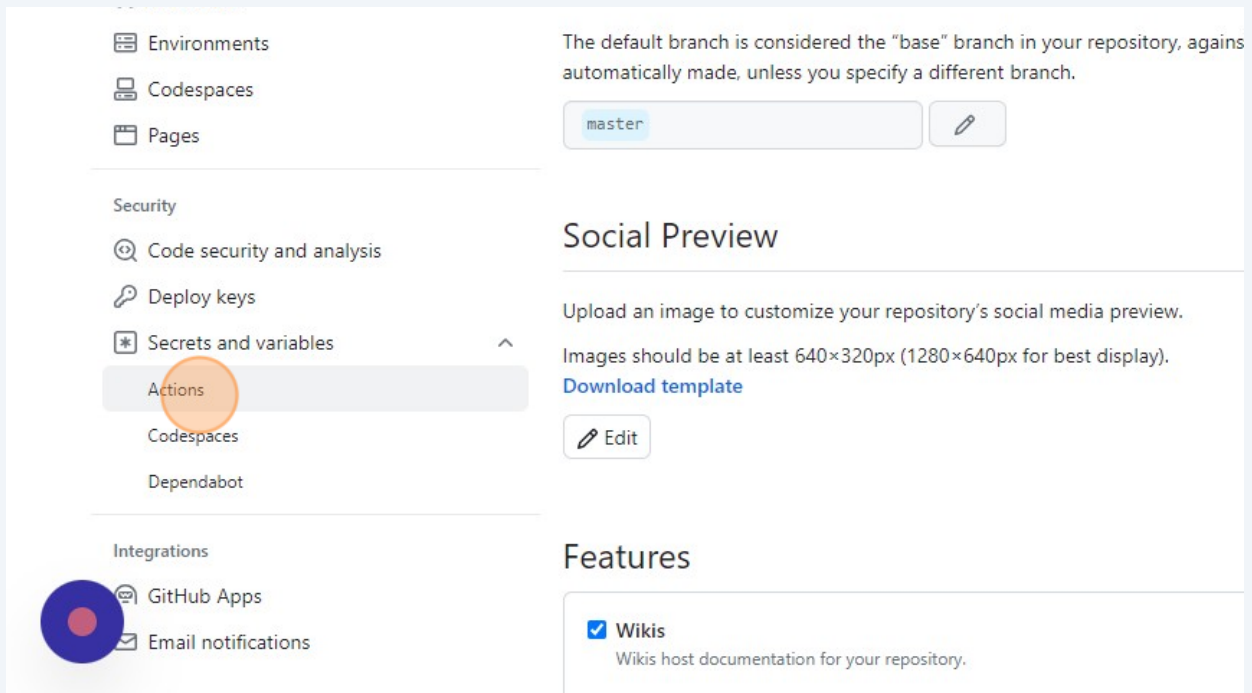
Edit

Features

☒ Wikis

Wikis host documentation for your repository.

4 Click "Actions"



The screenshot shows the GitHub repository settings page. On the left sidebar, under the 'Security' section, 'Actions' is highlighted with an orange circle. The main content area shows the 'master' branch selected in a dropdown menu. Below this, there are sections for 'Social Preview' and 'Features'. The 'Social Preview' section includes a description, a 'Download template' link, and an 'Edit' button. The 'Features' section shows the 'Wikis' feature is enabled.

Environments

Codespaces

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Actions

Codespaces

Dependabot

Integrations

GitHub Apps

Email notifications

The default branch is considered the "base" branch in your repository, against automatically made, unless you specify a different branch.

master

Social Preview

Upload an image to customize your repository's social media preview. Images should be at least 640×320px (1280×640px for best display). [Download template](#)

Edit

Features

☒ Wikis

Wikis host documentation for your repository.

5 Click "New repository secret"

Secrets allow you to manage reusable configuration data. Secrets are encrypted and are used for sensitive data. Variables are shown as plain text and are used for non-sensitive data. [Learn more](#)

Anyone with repository access to this repository can use these secrets and variables for actions. They are not passed to forks. They are not passed to pull requests triggered by a pull request from a fork.

Variables

New repository secret

Environment secrets

Manage environments

There are no secrets for this repository's environments.

Repository secrets

There are no secrets for this repository.

6 Click the "Name" field.

thomasjv799 / react-sample-app

Q Type to search

> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

General

Access

Collaborators

Moderation options

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Actions secrets / New secret

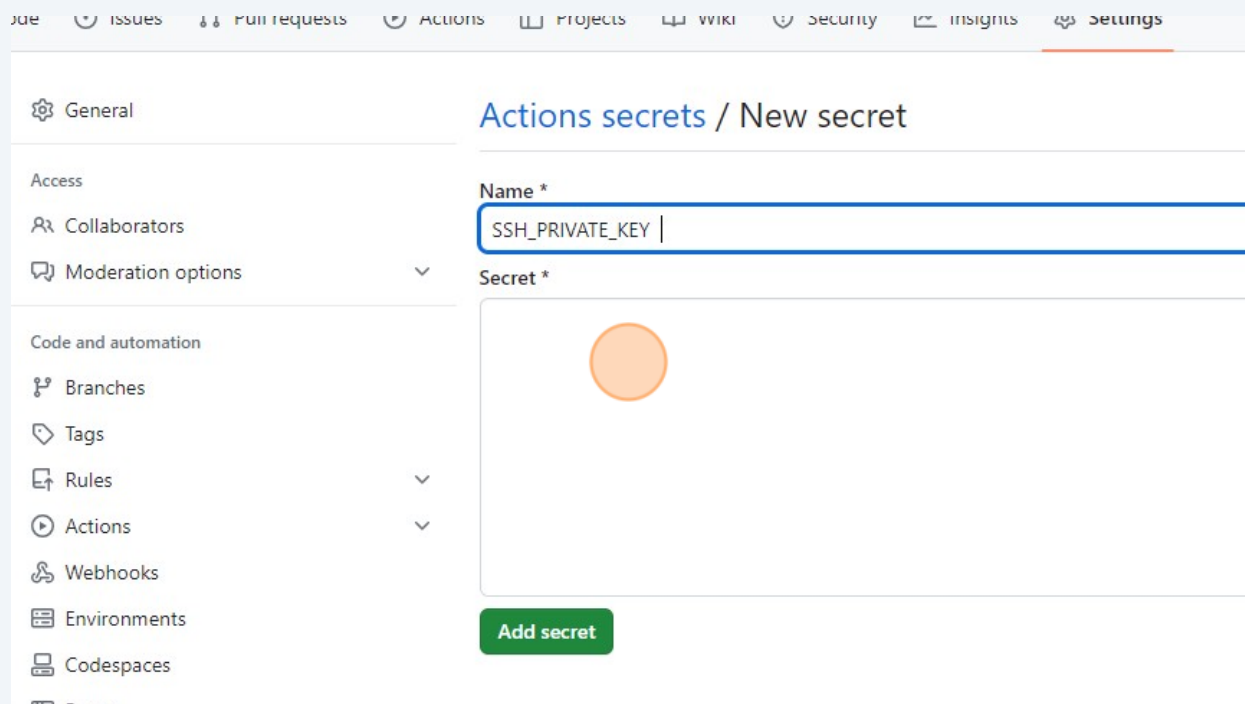
Name *

YOUR_SECRET_NAME

Secret *

7

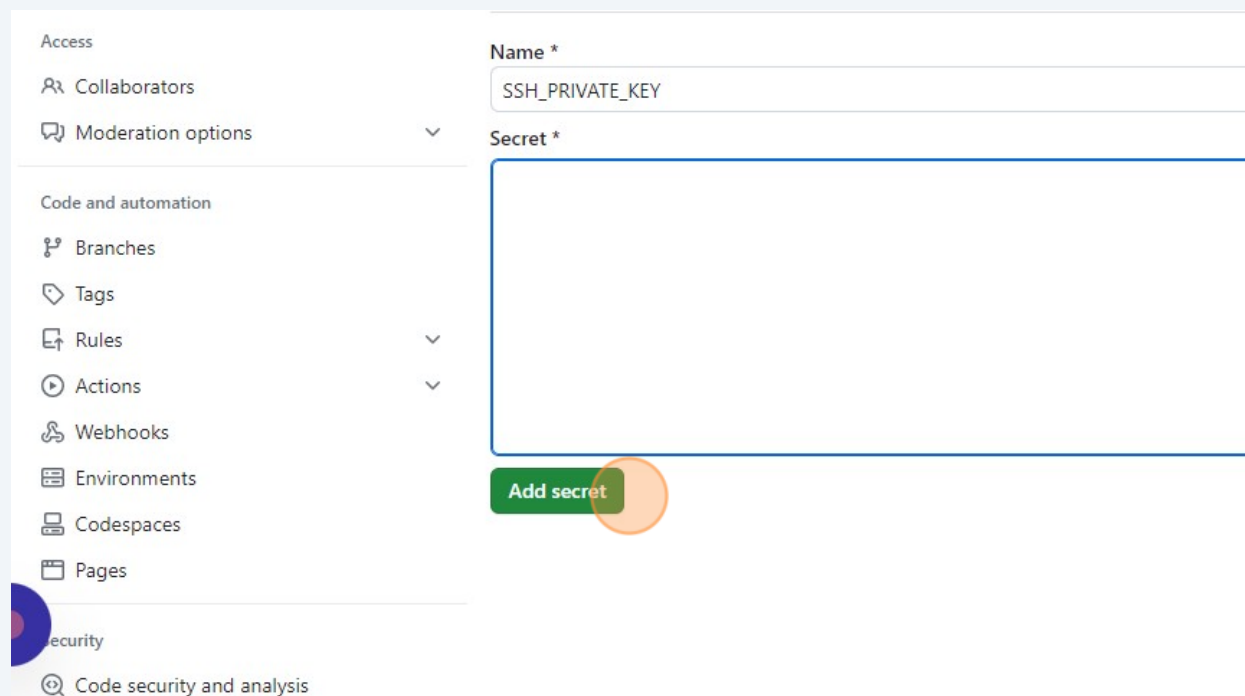
Click the "Secret
*" field.



The screenshot shows the GitHub Actions 'Secrets' page. The left sidebar contains a navigation menu with categories: 'General' (containing 'Access', 'Collaborators', 'Moderation options'), 'Code and automation' (containing 'Branches', 'Tags', 'Rules', 'Actions', 'Webhooks', 'Environments', 'Codespaces'), and 'Security'. The 'Actions' item is selected. The main content area is titled 'Actions secrets / New secret'. It features two input fields: 'Name *' with the value 'SSH_PRIVATE_KEY' and 'Secret *' which is currently empty and has a large orange circle highlighting it. Below the 'Secret' field is a green 'Add secret' button.

8

Add the SSH Private key of the EC2 machine here.



This screenshot is similar to the previous one, showing the 'New secret' form. The 'Name' field contains 'SSH_PRIVATE_KEY'. The 'Secret' field is empty. In this step, a large orange circle highlights the green 'Add secret' button at the bottom of the form.

9

Add the same for HOSTNAME and USER_NAME which is the IP address of EC2 Machine and user name of the machine. Make sure that SSH key of the machine is added in the github.

thomasjv799 / react-sample-app

Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Actions secrets / New secret

Name *

SSH_PRIVATE_KEY

Secret *

10

Once all three secrets are added, go and click on actions.

Package a Jekyll site using the jekyll/builder Docker image.

Configure HTML

Publishes a Node.js package to npm.

Configure JavaScript

Gulp

By GitHub Actions

Build a NodeJS project with npm and gulp.

Configure JavaScript

Webpack

By GitHub Actions

Build a NodeJS project with npm and webpack.

Configure JavaScript

View all

Deploy to Azure

Deploy to Amazon ECS

By Amazon Web Services

Build and Deploy to GKE

By Google Cloud

11 There, Click "set up a workflow yourself"

Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select this and [set up a workflow yourself](#) →

Suggested for this repository

Publish Node.js Package to GitHub Packages
By GitHub Actions
Publishes a Node.js package to GitHub Packages.
Configure
JavaScript

Jekyll using Docker image
By GitHub Actions
Package a Jekyll site using the jekyll/builder Docker image.
Configure
HTML

12 We need to add the yaml content here, I have put a sample on the folder where we directly can copy and paste.

thomasjv799 / react-sample-app

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

react-sample-app / .github / workflows / main.yml in master

Edit Preview Spaces 2 No wrap

1 Enter file contents here

13 Click "sudo systemctl restart apache2"

```
7
8 jobs:
9   Deploy:
10     runs-on: ubuntu-latest
11
12     steps:
13       - uses: actions/checkout@v2
14       - name: Build & Deploy
15         env:
16           PRIVATE_KEY: ${ secrets.SSH_PRIVATE_KEY }
17           HOSTNAME: ${ secrets.HOSTNAME }
18           USER_NAME: ${ secrets.USER_NAME }
19
20         run: |
21           echo "$PRIVATE_KEY" > private_key && chmod 600 private_key
22           ssh -o StrictHostKeyChecking=no -i private_key ${USER_NAME}@${HOSTNAME} '
23             cd /home/ubuntu/reactjs-sample-app &&
24             sudo npm run build &&
25             echo "Sucessfully Build" &&
26             cp -r build /var/www &&
27             cd /etc/apache2/sites-available &&
28             sudo a2ensite 001-reactjs-prod.conf &&
29             sudo systemctl restart apache2
30           '
```



Cache

By actions ✓

Cache artifacts like dependencies and outputs to improve workflow execution



Setup Node.js environment

By actions ✓

Setup a Node.js environment by installing the required dependencies, optionally downloading and adding it to the path



Setup Java JDK

By actions ✓

14 These are the steps that we are gonna do inside the EC2 machine, we can modify according to the requirement.

> src

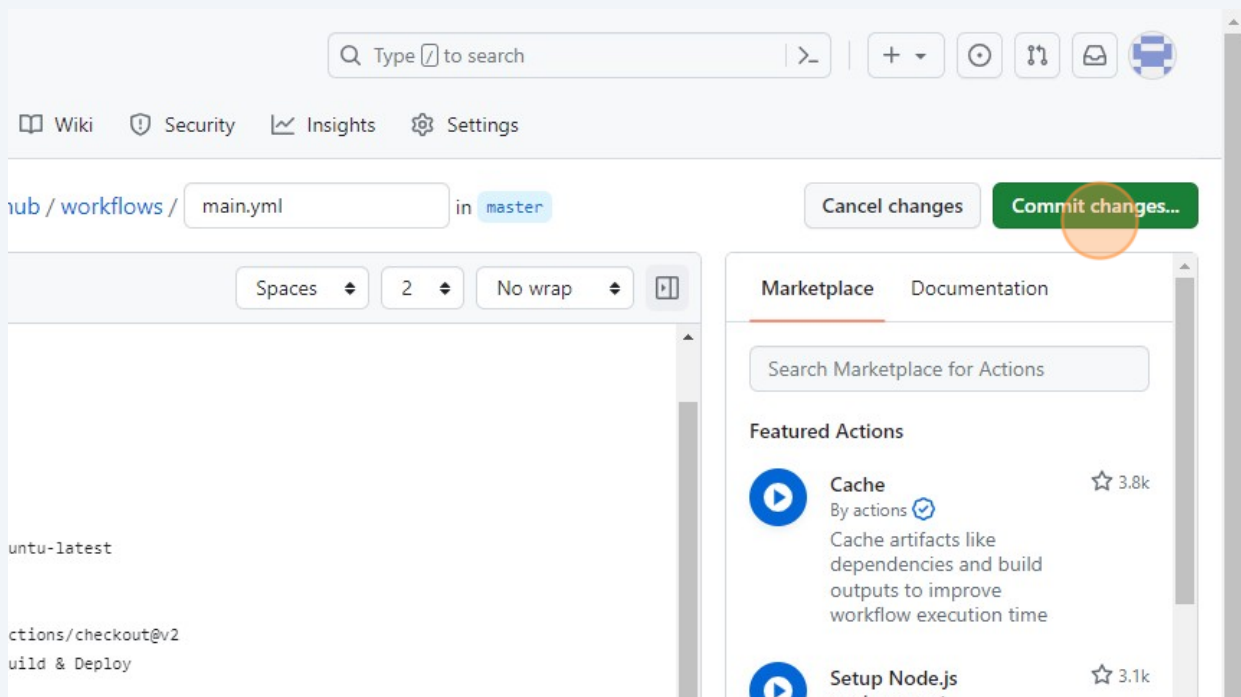
- .gitignore
- README.md
- package-lock.json
- package.json



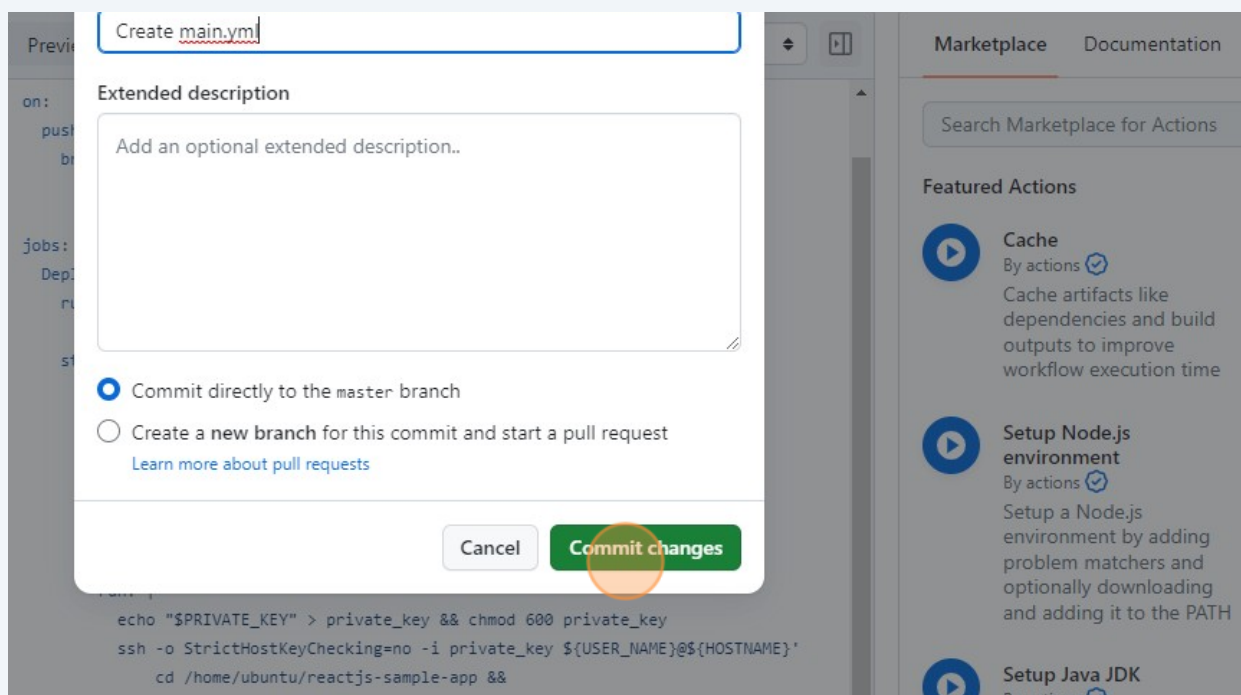
Documentation • Share feedback

```
7
8 jobs:
9   Deploy:
10     runs-on: ubuntu-latest
11
12     steps:
13       - uses: actions/checkout@v2
14       - name: Build & Deploy
15         env:
16           PRIVATE_KEY: ${ secrets.SSH_PRIVATE_KEY }
17           HOSTNAME: ${ secrets.HOSTNAME }
18           USER_NAME: ${ secrets.USER_NAME }
19
20         run: |
21           echo "$PRIVATE_KEY" > private_key && chmod 600 private_key
22           ssh -o StrictHostKeyChecking=no -i private_key ${USER_NAME}@${HOSTNAME} '
23             cd /home/ubuntu/reactjs-sample-app &&
24             sudo npm run build &&
25             echo "Sucessfully Build" &&
26             cp -r build /var/www &&
27             cd /etc/apache2/sites-available &&
28             sudo a2ensite 001-reactjs-prod.conf &&
29             sudo systemctl restart apache2
30           '
```


15 Click "Commit changes..."



16 Click "Commit changes"



17 Click "Actions"

The screenshot shows the GitHub repository page for 'thomasjv799 / react-sample-app'. The 'Actions' tab is highlighted with an orange circle. The left sidebar shows the file explorer with the following structure:

- Code
 - master
 - Go to file
 - .github/workflows
 - main.yml
 - public
 - src
 - .gitignore
 - README.md
 - package-lock.json

The main content area shows the 'react-sample-app / .github / workflows /' path. Below this, there is a table with the following content:

Name	Last commit message
..	
main.yml	Create main.yml

18 Click "Deploy to EC2 Instance"

The screenshot shows the GitHub repository page for 'thomasjv799 / react-sample-app'. The 'Actions' tab is selected. The left sidebar shows the 'Actions' section with the following items:

- All workflows
- Deploy to EC2 Instance (highlighted with an orange circle)
- Management
- Caches

The main content area shows the 'All workflows' section with the text 'Showing runs from all workflows'. Below this, there is a section titled '0 workflow runs' with a blue play button icon and the text 'There are no worl'.

19 Our Github Pipeline is ready, now if we push anything it will trigger the pipeline.