

Gitea and DroneCI

I use [Gitea](#) to self-host my code + projects [here](#) and I use [DroneCI](#), a lightweight CI pipeline that integrates into gitea to do automation stuff.

Configuration of Drone + Gitea

TODO: write about setup here - docker etc

Drone CI Config

Drone works like many other CI systems - use yaml files in the repository to control builds. You can specify the type of CI run at the top level of the yaml doc and also give it a name:

```
kind: pipeline
type: docker
name: test and build
```

You can define multiple build steps - each one can have its own docker image - this is useful for example if you have a React frontend and a Golang backend and you need to build both

```
steps:
  - name: test_backend
    image: python:3.7
    commands:
      - pip install poetry
      - poetry install
      - poetry run pytest

  - name: test_frontend
    image: node
    commands:
      - npm install
      - npm test
```

Shared State Between Steps

Steps can share files via [temporary volumes](#) if needed but are generally stateless and independent.

Conditional Execution of Steps

You can opt to only run steps under certain conditions. For example you might only want to publish code when stuff gets pushed to master. Use conditions to do this:

```
- name: publish
  when:
    branch:
      - master
    event:
      exclude:
        - pull_request
  image: python3.7
  commands:
    - twine upload...
```

[The drone documentation](#) has a full set of conditions that you can use - you can also whitelist and blacklist certain events and certain branches

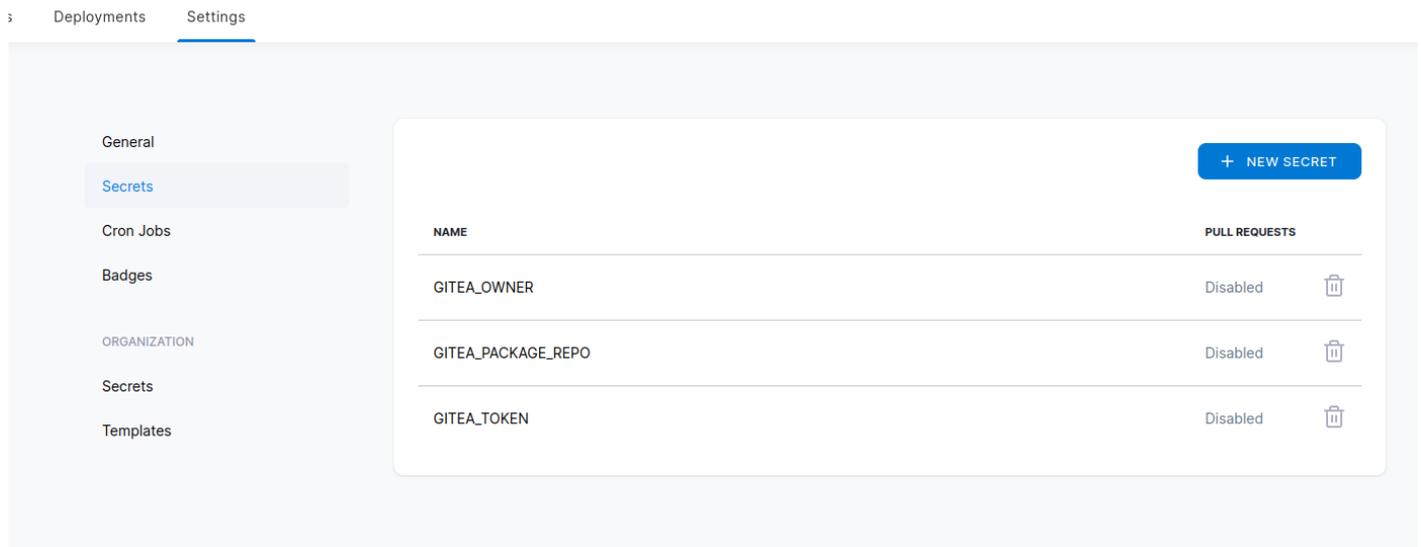
Secrets

[Secrets](#) can be used to pass things like auth tokens into CI pipelines. This is useful if you want to do things like publish packages or upload files. You essentially declare an environment variable and then define which "secret" it came from in your CI yaml file:

```
- name: publish
  when:
    branch:
      - master
    event:
      exclude:
        - pull_request
  image: python3.7
  environment:
```

```
GITEA_PACKAGE_REPO:  
  from_secret: gitea_package_repo  
GITEA_OWNER:  
  from_secret: gitea_owner  
GITEA_TOKEN:  
  from_secret: gitea_token
```

Then in the drone frontend you can add the value and it will be stored securely and passed to the CI at run time:



The screenshot shows the Drone CI web interface. At the top, there are tabs for 'Deployments' and 'Settings'. The 'Settings' tab is active. On the left side, there is a sidebar menu with options: 'General', 'Secrets' (highlighted), 'Cron Jobs', 'Badges', 'ORGANIZATION', 'Secrets', and 'Templates'. The main content area displays a table of secrets. At the top right of this area is a blue button labeled '+ NEW SECRET'. The table has two columns: 'NAME' and 'PULL REQUESTS'. There are three rows of secrets listed:

NAME	PULL REQUESTS
GITEA_OWNER	Disabled 
GITEA_PACKAGE_REPO	Disabled 
GITEA_TOKEN	Disabled 

Backup Mechanism

Revision #5

Created 22 October 2022 11:09:34 by James

Updated 21 January 2024 14:54:00 by James