automated deployments from github

with circleci & ansible

https://github.com/shortstack/circleci-demo-ermahgerd


AUTOMATION
SAN DIEGO
IT'S KIND OF A BIG DEAL
memegenerator.net

the plan

1. push to github
2. build in circleci
3. deploy via circleci + awscli *OR* circleci + ansible

srsly -->

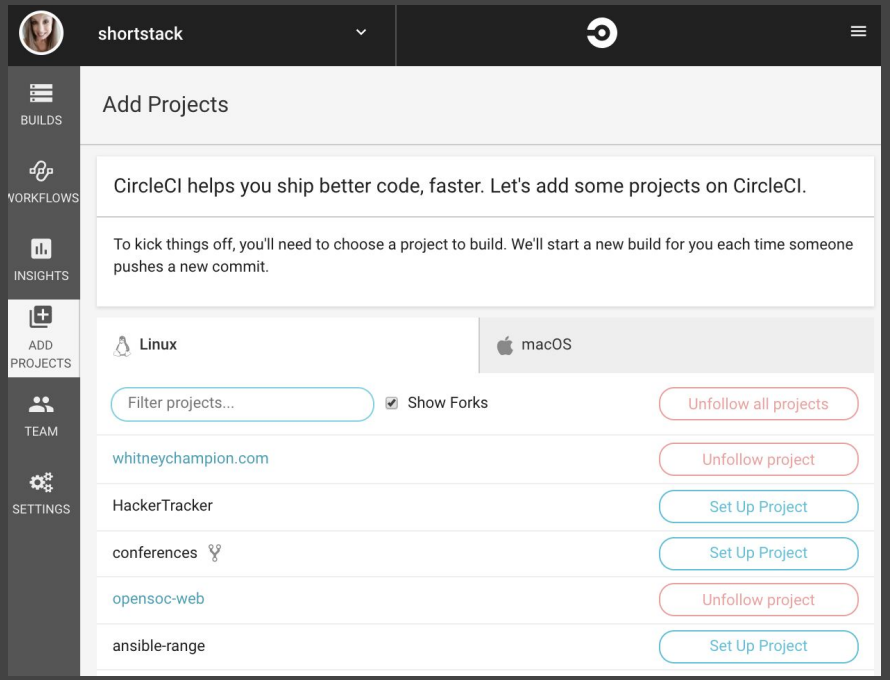
things you'll need

- github account
- static content in a repo in said github account
- circleci account (can auth with github)
- AWS access keys
  - if deploying to AWS*
- SSH key pair
  - if deploying to a server*

pick one

set up circleci

- log into circleci → add projects → set up project
- pick OS and language
- make a .circleci directory in your repo
- touch .circleci/config.yml
- git commit -a
- git push origin master
- click start building
- FYI, this will probably fail
- BUT THAT'S OK! \o/



if using S3...

configure aws

- if you already have access keys with at least S3 permissions, skip this
- if you don’t…
  - log into AWS IAM → users
  - add user
  - name + programmatic access
  - permissions → “attach existing policies”
    - S3 full access
  - create user
  - download keys, keep somewhere safe

configure aws in circleci

- project settings
- AWS permissions
- add access keys

deploying to S3

- make an S3 bucket
- properties → static website → use this to host a website

deploying to S3

config.yml

```yaml
version: 2
jobs:
  deploy:
    docker:
      - image: circleci/python:2.7
    working_directory: ~/circleci-demo-ermahgerd
    steps:
      - checkout
      - run:
          name: install awscli
          command: sudo pip install awscli
      - run:
          name: remove .git and .circleci
          command: rm -rf .git && rm -rf .gitignore && rm -rf .circleci
      - run:
          name: deploy to S3
          command: aws s3 sync . s3://ermahgerd --region us-east-1 --acl public-read
workflows:
  version: 2
  build-deploy:
    jobs:
      - deploy:
          filters:
            branches:
              only: master
```

if using your own server instead of S3...

add user & ssh key

- ssh you@server

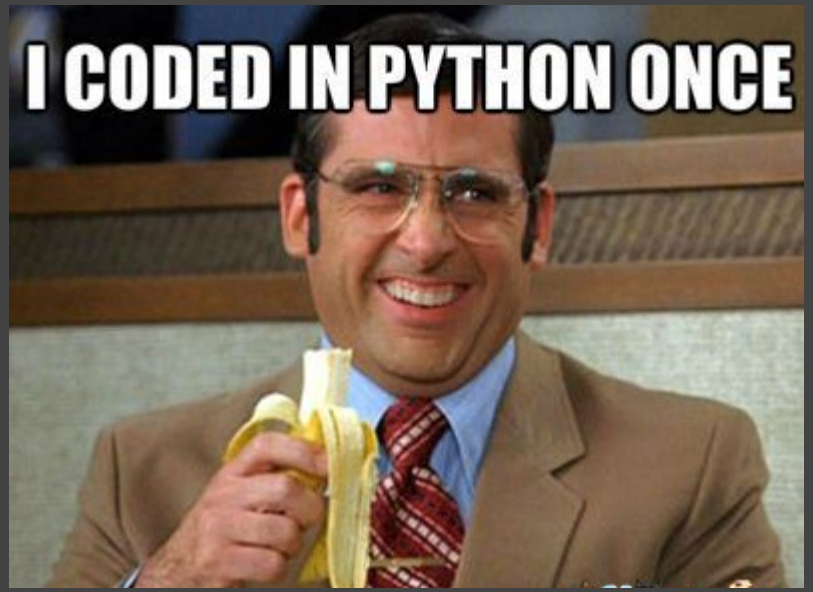
```
sudo useradd circleci
sudo chown circleci:circleci /var/www/html
su - circleci
ssh-keygen -t rsa -b 4096
cat ~/.ssh/id_rsa.pub > ~/.ssh/authorized_keys
chmod 644 ~/.ssh/authorized_keys
```

configure ssh

- circleci → project settings
- SSH permissions
- add SSH key
  - set hostname
  - paste private key (.ssh/id_rsa contents)

install python

- sudo apt or yum install python -y

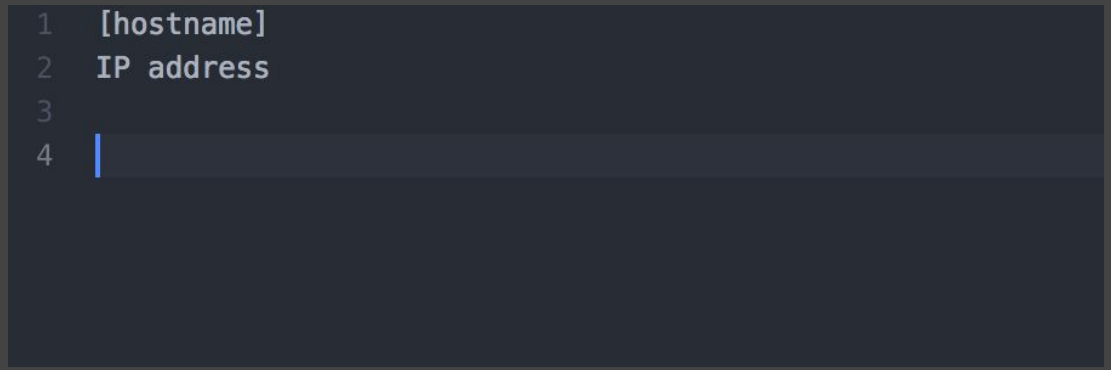


environment variables

- circleci → project settings
- environment variables
- add variable
  - ANSIBLE_HOSTS = ~/repo-name/.circleci/hosts
  - ANSIBLE_HOST_KEY_CHECKING = False

configure ansible

- this is your ansible inventory
- .circleci/hosts



deploying to a server

- this assumes you already have a web server running
  - install web server package(s)
  - start and enable web server
  - locate default directory to put content in
  - chown circleci:circleci /directory

deploying to a server

config.yml

```yaml
version: 2
jobs:
  deploy:
    docker:
      - image: circleci/python:2.7
    working_directory: ~/circleci-demo-ermahgerd
    steps:
      - checkout
      - run:
          name: install ansible
          command: sudo pip install ansible
      - run:
          name: install rsync
          command: sudo apt install -y rsync
      - run:
          name: deploy via ansible
          command: ansible-playbook .circleci/deploy.yml

workflows:
  version: 2
  build-deploy:
    jobs:
      - deploy:
          filters:
            branches:
              only: master
```

ansible playbook - deploy.yml

```yaml
---
- hosts: 167.99.148.100
  become: yes
  tasks:

    - name: rsync files
      synchronize:
        src: ~/circleci-demo-ermahgerd/
        dest: /var/www/html/
        rsync_opts:
          - "--exclude=.git --exclude=.circleci"

```

now what?!

let’s test it out

- git commit & git push
- go to circleci dashboard
- SEE WHAT HAPPENS

the end <3