BUILD AND RELEASE CONFIDENTLY WITH CONTINUOUS INTEGRATION AND DELIVERY

Steve Grunwell
@stevegrunwell

stevegrunwell.com/slides/intro-to-ci-cd

WHAT IS CI/CD?

CONTINUOUS INTEGRATION

- Run a series of scripts automatically, any time changes are pushed
- **Continuously integrate** our changes

CONTINUOUS INTEGRATION

- Automated tests
- Coding standards
- Static code analysis
- ...etc.

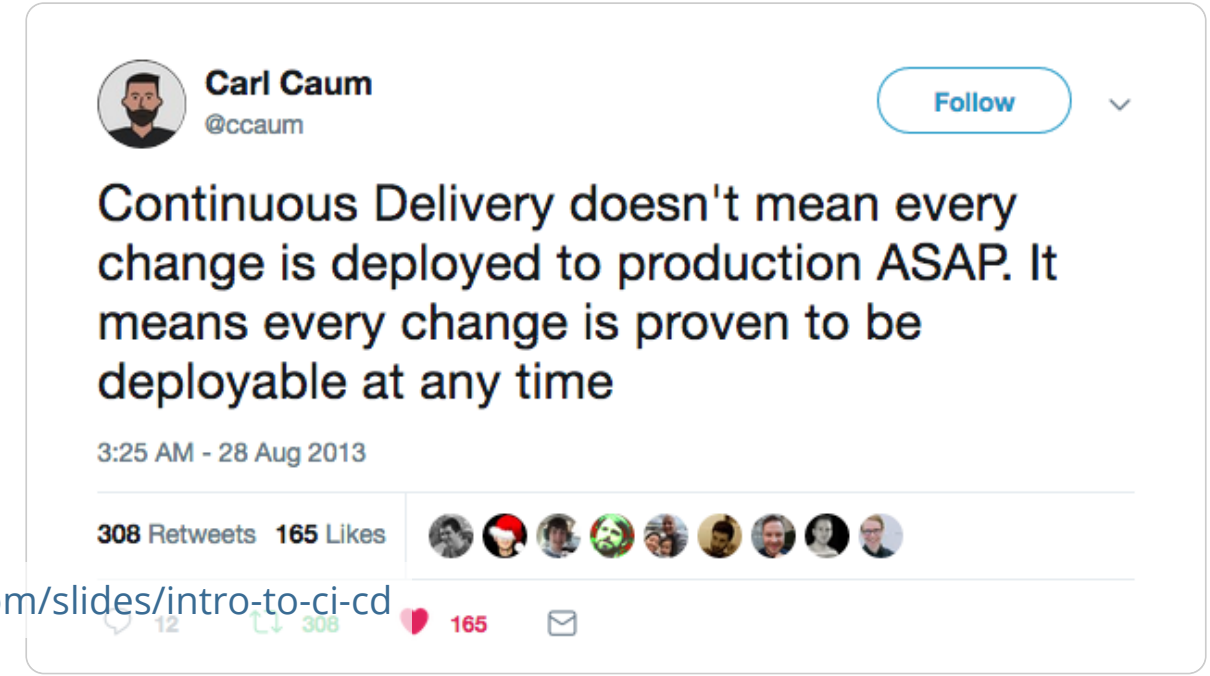
DEPENDENCIES & ARTIFACTS

- Dependency management
- Anything that's generated for your app
  - Compiled and/or minified files
  - Binaries
- Source control should only worry about the source

CONTINUOUS DELIVERY

- Being able to be deploy on-demand
- One-click deployments

DELIVERY vs. DEPLOYMENT



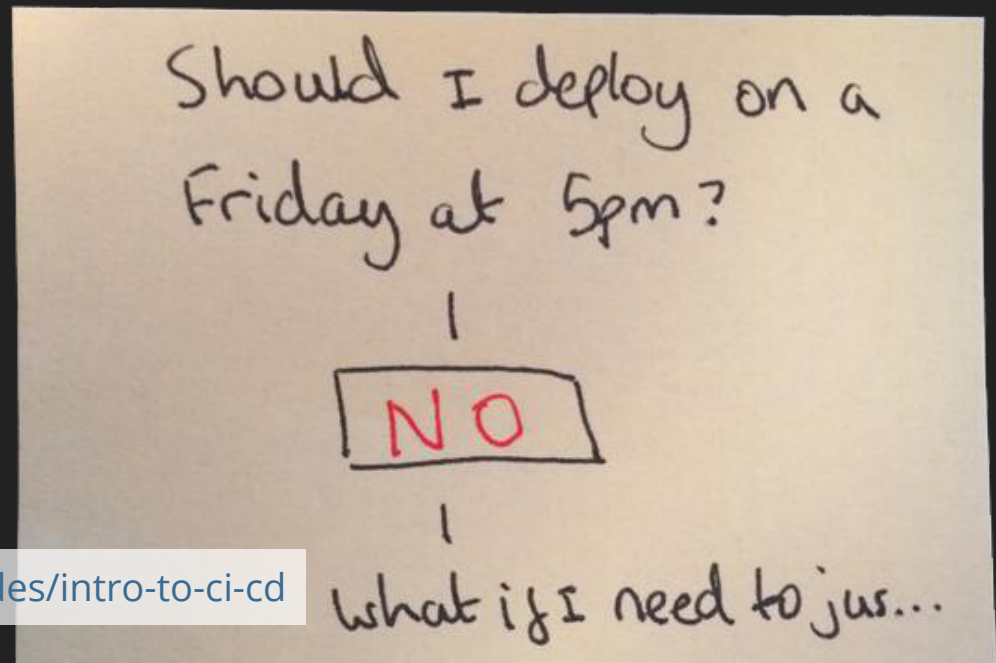
DELIVERY vs. DEPLOYMENT

Delivery

Some manual step to deploy

Deployment

Always Be Deployin'


Should I deploy on a Friday at 5pm?
NO
what if I need to jus...

DEPLOY WITH CONFIDENCE!

In order to deploy confidently,
we must have confidence in our tools.

SETTING UP A CI/CD PIPELINE

WHAT IS A PIPELINE?

- A route from development to production
- Different branches may take different paths

CI/CD PROVIDERS

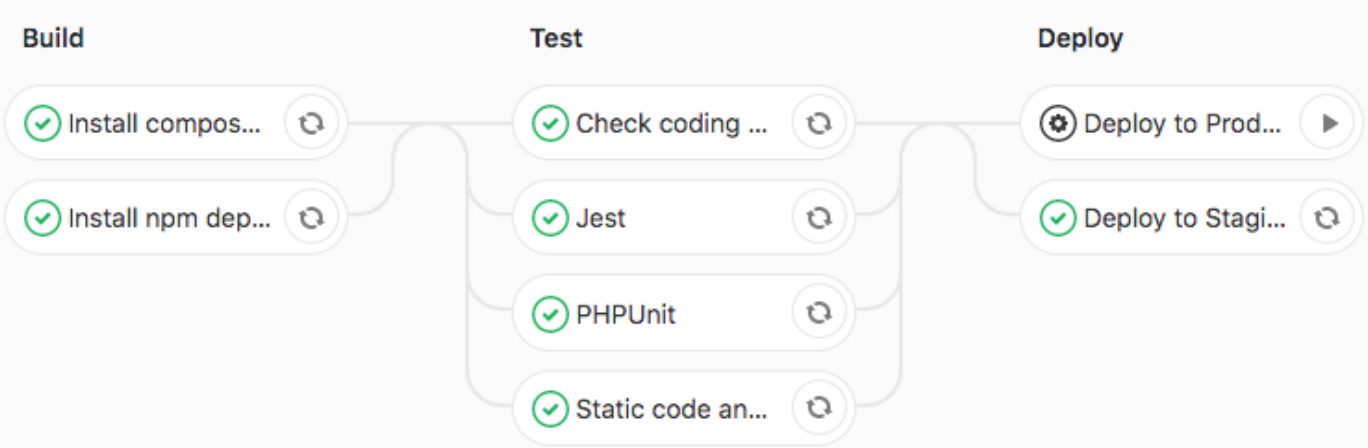
- Jenkins
- Travis CI
- Circle CI
- Codeship
- DeployBot

- Source code management tool
- Hosted platform or available for private installation
- Includes CI/CD tools!

GITLAB CI/CD TOOLS

- Define multiple **stages** (build, test, deploy, etc.)
- Each stage has one or more **jobs**
- Multiple runners == run jobs in parallel!

A TYPICAL PIPELINE



PIPELINE CONFIGURATION

`.gitlab-ci.yml`

A SIMPLE JOB

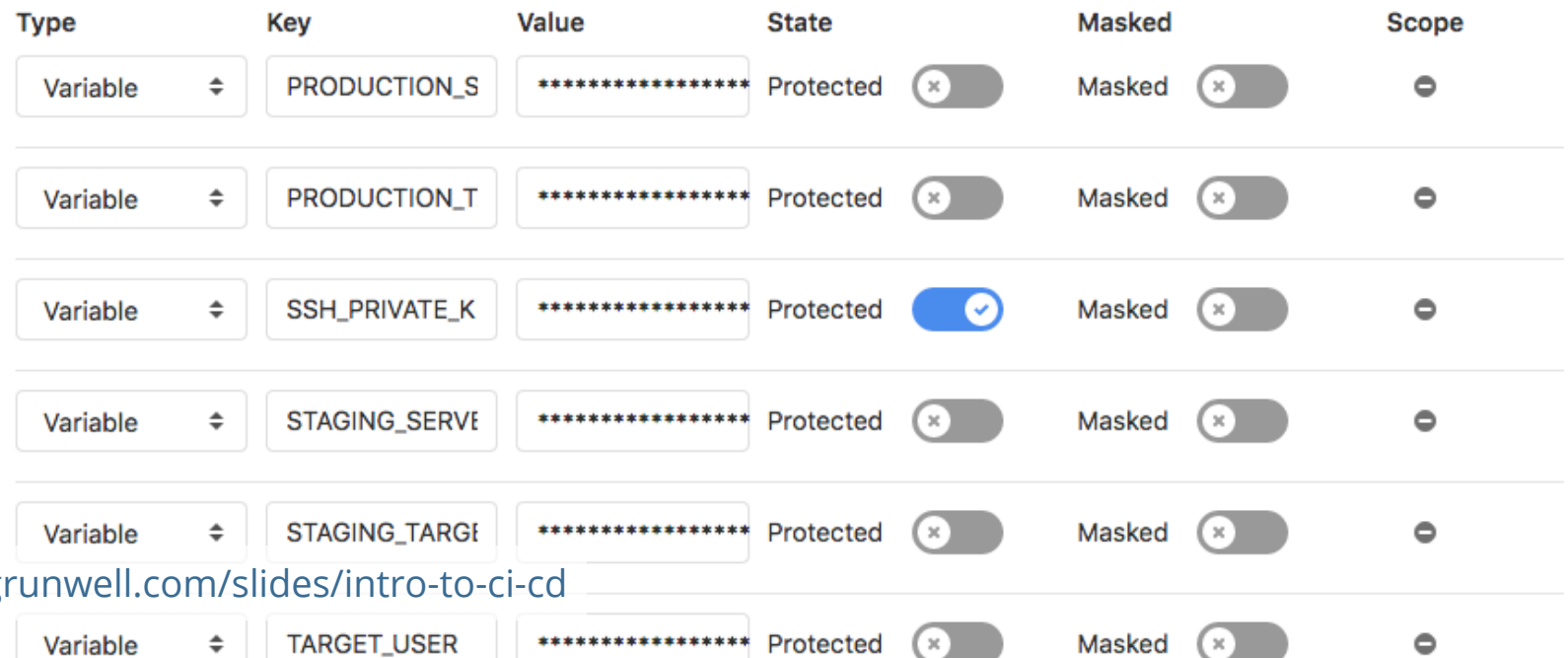
```
Install npm dependencies:
    stage: build
    script:
        - npm install --no-progress
        - npm run prod

    artifacts:
        paths:
            - public
    cache:
        key: ${CI_COMMIT_REF_SLUG}-npm
        paths:
            - node_modules
```

...AND SO MUCH MORE!

- `image`
- `only` + `except`
- `dependencies`

docs.gitlab.com/ee/ci/yaml

ENVIRONMENT VARIABLES



SIMPLE DEPLOYMENTS

WHAT DOES OUR DEPLOYMENT PROCESS LOOK LIKE RIGHT NOW?

- (S)FTP?
- SSH + `git pull`?
- Docker?

A DROP-DEAD SIMPLE DEPLOYMENT:

1. Build the app
2. scp a tarball to production
3. rsync the files into the web root

A DROP-DEAD SIMPLE DEPLOYMENT:

```
ship_to_production:
    stage: deploy
    script:
        - tar -czf release.tgz dist/*

        - scp release.tgz "${SSH_USER}@${PRODUCTION_ADDR}:/tmp"
        - ssh "${SSH_USER}@${PRODUCTION_ADDR}" "cd /tmp \
          && tar -xzf release.tgz \
          && rsync release/ /var/www/html/ \
          "
```

ATOMIC DEPLOYMENTS

releases/

Contains multiple, timestamped deployments

shared/

Data that should persist between deployments

current

Symlink to the current release in `releases/`

ATOMIC DEPLOYMENTS

```
# /var/www
releases/
  + 1563759801/ # Oldest
  + 1563759802/
  + 1563759803/ # Newest
shared/
  + logs/
  - config.json
```

```
current => /var/www/releases/1563759802 # Point to the previous
```

ATOMIC DEPLOYMENTS

```
ship_to_production:
    stage: deploy
    script:
        # 1. Create + scp tarball
        # 2. Extract to /var/www/releases/{TIMESTAMP}

        # 3. Symlink shared assets
        # 4. Update the `current` symlink
        # 5. Reload the web server
        # 6. (Optional) Roll off older releases
```

stevegrunwell.com/blog/atomic-deployments-from-scratch

EXTENDING OUR PIPELINES

BLUE-GREEN DEPLOYMENTS

- Run multiple production environments/servers
  - Some active (blue), some idle (green)
- Deploy to green
- Once ready, route traffic to green
  - Blue becomes idle
- In case of issues, re-route to blue

BLUE-GREEN DEPLOYMENTS

BUILDING A DOCKER IMAGE

- Final product: a Docker image of your app
- Great for Docker-powered production clusters!

GENERATE DOCUMENTATION

Automatically parse + release documentation!

CODE COVERAGE REPORTS

- Generate code coverage as part of the CI pipeline
- Help identify branches that reduce code coverage

NOTIFICATIONS

- GitLab CI/CD supports webhooks
- Can be used to post to Slack, send emails, and more!

REMEMBER: THIS ISN'T MAGIC!

THANK YOU!

Steve Grunwell
Senior Software Engineer, Liquid Web

stevegrunwell.com/slides/intro-to-ci-cd