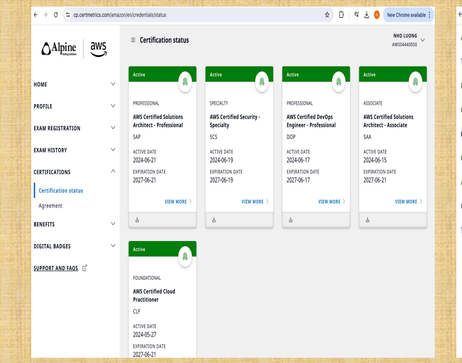
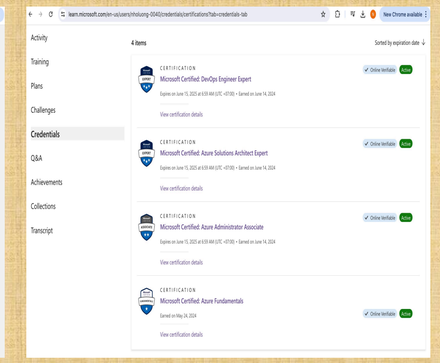
Build & Release Confidently with Continuous Integration & Delivery

Author: Nho Luong
Skill: DevOps Engineer Lead





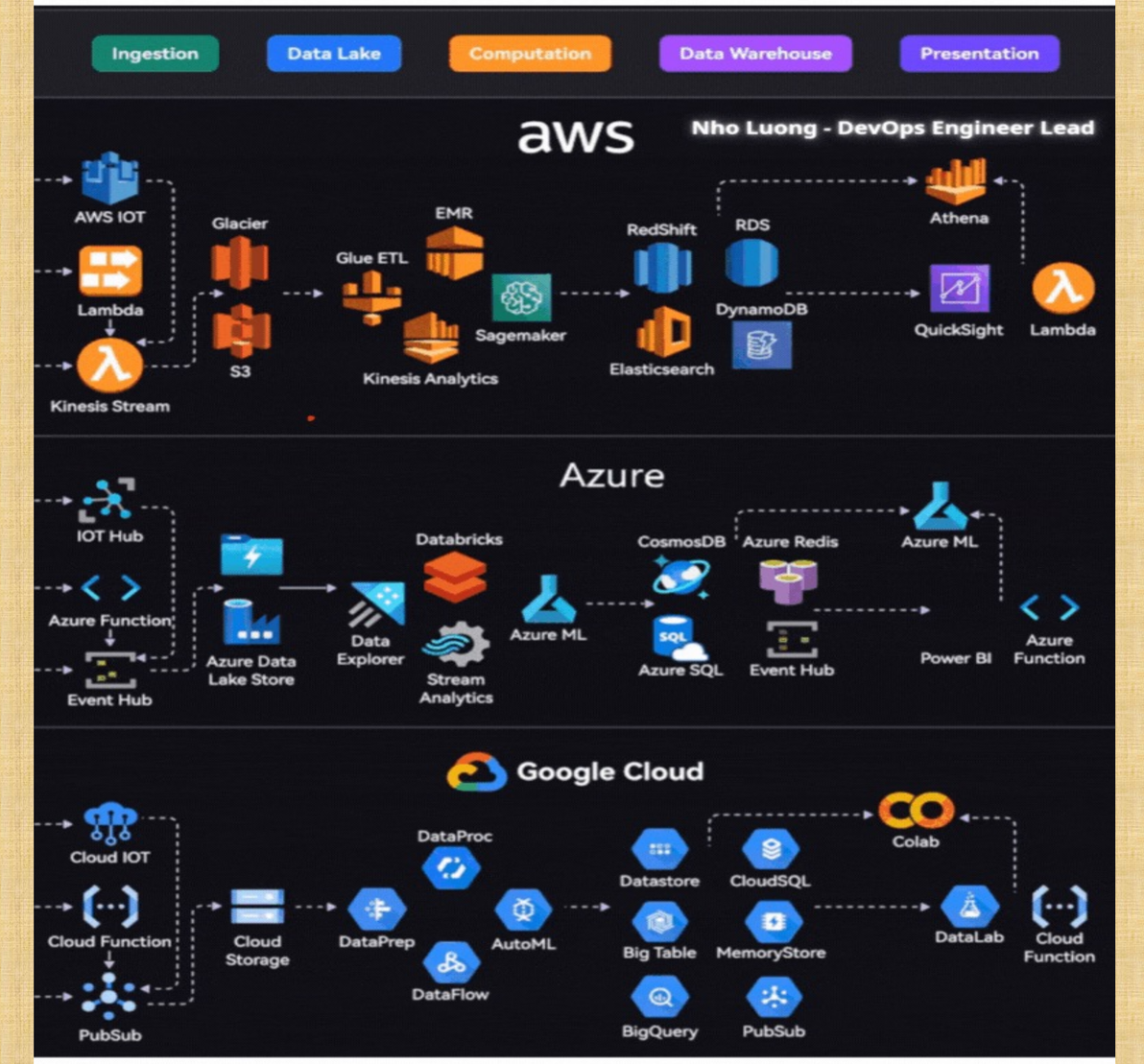
What is CI/CD?

Author: Nho Luong
Skill: DevOps Engineer Lead

Continuous Integration

- Run a series of scripts automatically, any time changes are
- ~~pushed~~ **Continuously Integrate** our changes Automated
- tests Coding standards Static code analysis ...et cetera
-
-
-

Author: Nho Luong
Skill: DevOps Engineer Lead



Author: Nho Luong
Skill: DevOps Engineer Lead

Dependencies & Artifacts

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

Author: Nho Luong
Skill: DevOps Engineer Lead

Continuous Delivery

- Being able to deploy on-
- demand One-click
- deployments You can't forget a

  step if
  everything's automatic!

Author: Nho Luong
Skill: DevOps Engineer Lead

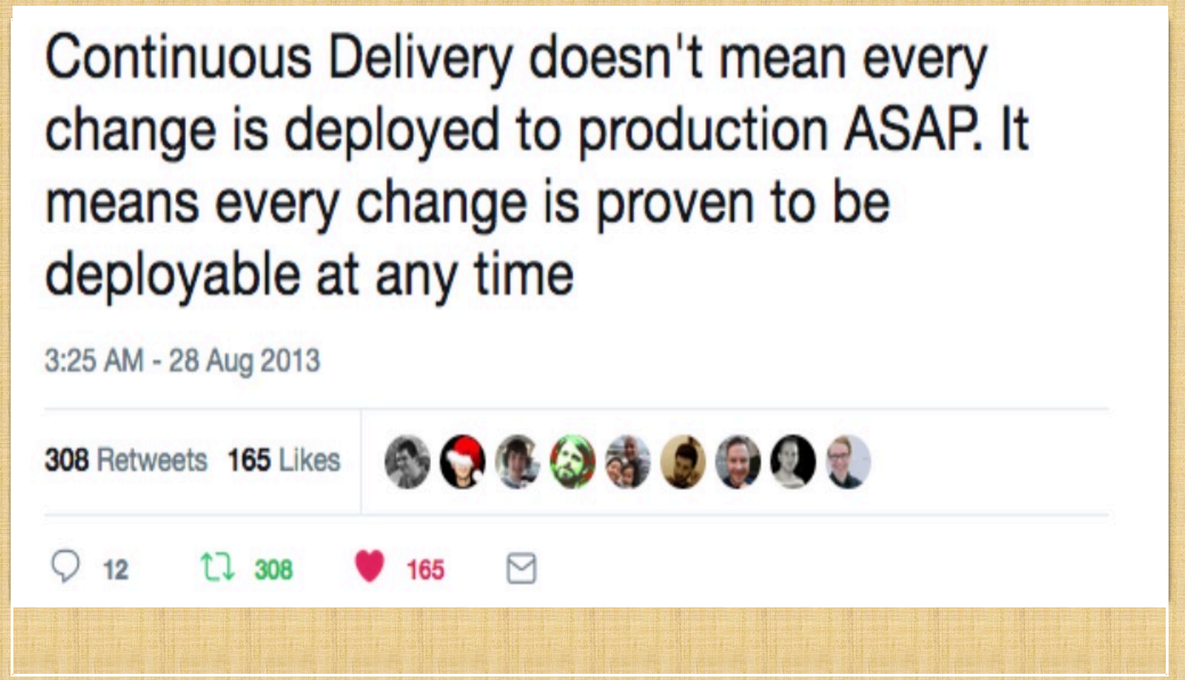
Continuous Delivery v Deployment

Delivery

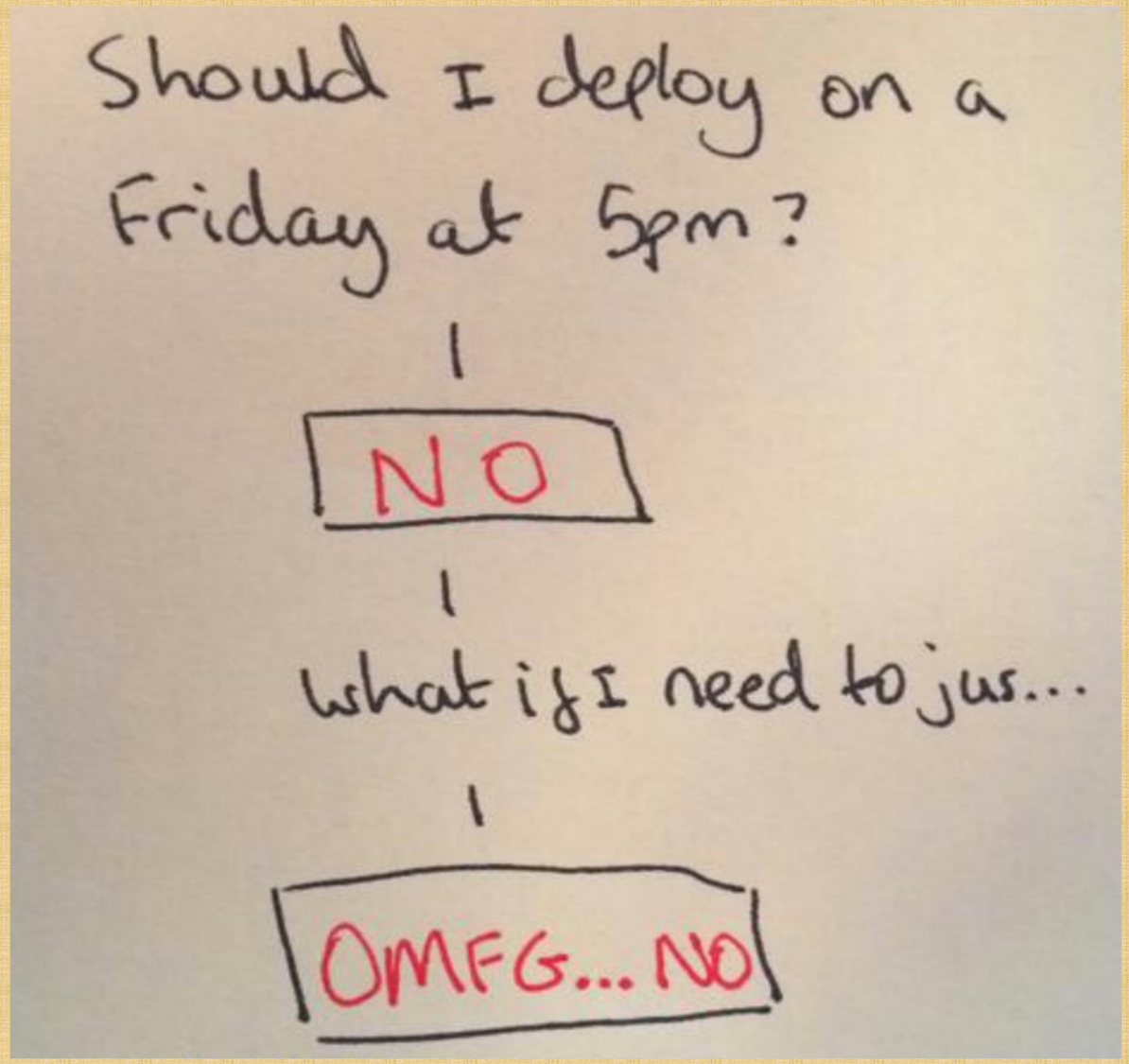
Some manual step to deploy

Deployment

Always Be Deployin'



Author: Nho Luong
Skill: DevOps Engineer Lead



Author: Nho Luong
Skill: DevOps Engineer Lead

Deploy with Confidence!

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

Author: Nho Luong
Skill: DevOps Engineer Lead

Setting up a CI/CD Pipeline

Author: Nho Luong
Skill: DevOps Engineer Lead

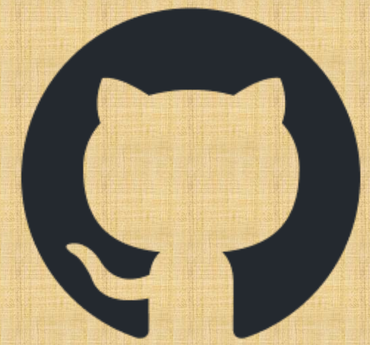
What is a Pipeline?

- A route from development to production
- Triggered by different events (push, merge, etc.)
- Different branches may take different paths

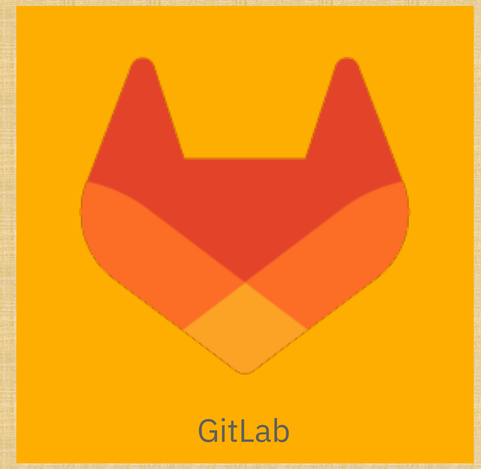
Photo by Christian Bass

Author: Nho Luong
Skill: DevOps Engineer Lead

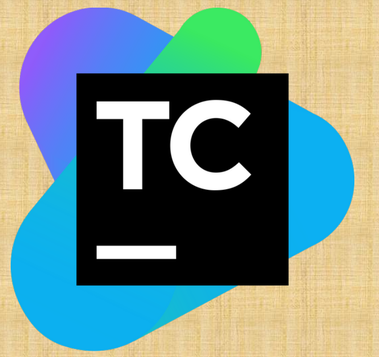
Popular CI/CD Providers



GitHub

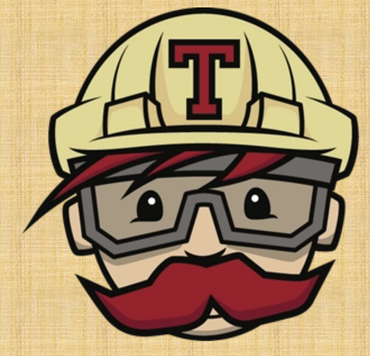


GitLab

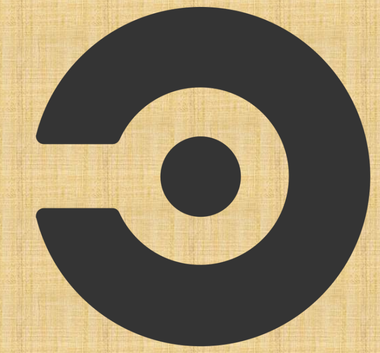


TeamCity

Jenkins

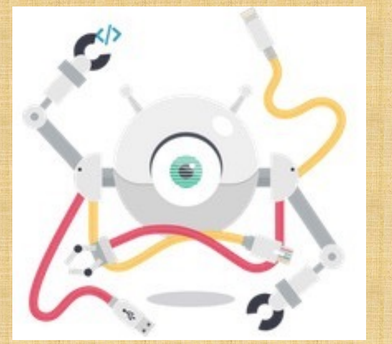


Travis CI



CircleCI

Codeship



Deploybot

Author: Nho Luong
Skill: DevOps Engineer Lead

Welcome to GitLab!

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

Author: Nho Luong
Skill: DevOps Engineer Lead

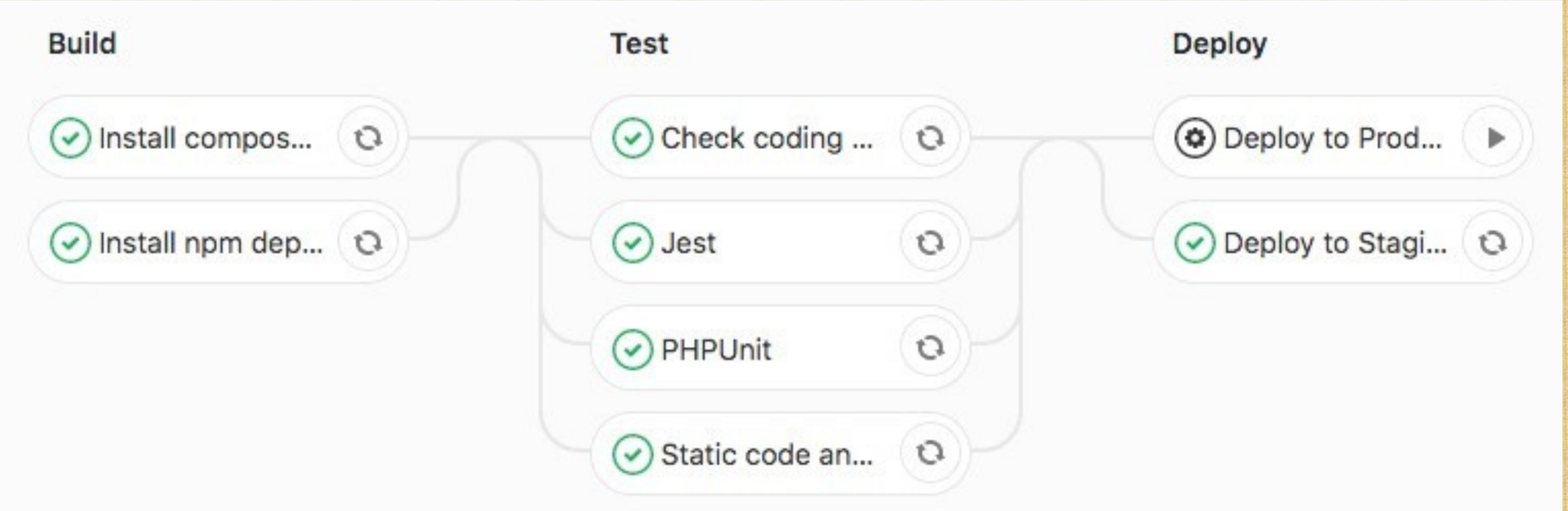
GitLab CI/CD Tools

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

Author: Nho Luong
Skill: DevOps Engineer Lead

A Typical Pipeline

Configured in .gitlab-ci.yml



Author: Nho Luong
Skill: DevOps Engineer Lead

A Simple Job

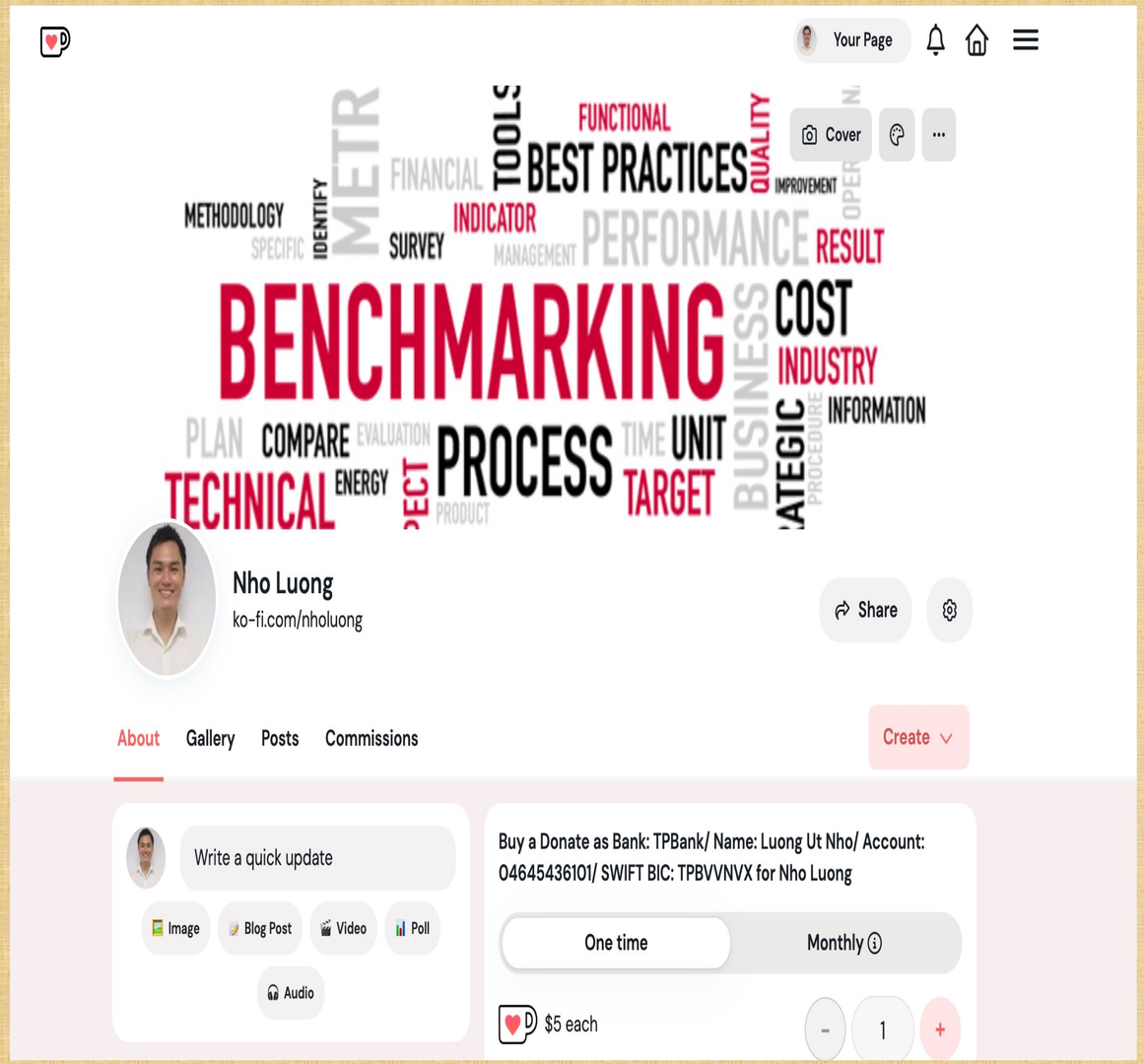
Install npm dependencies:

```
stage: build

script:
    - npm install --no-progress
    - npm run prod

artifacts:
  paths:
    - public

cache:
    key:
        files:
            - package-lock.json
        paths:
            -
            node_modules
```



Author: Nho Luong
Skill: DevOps Engineer Lead

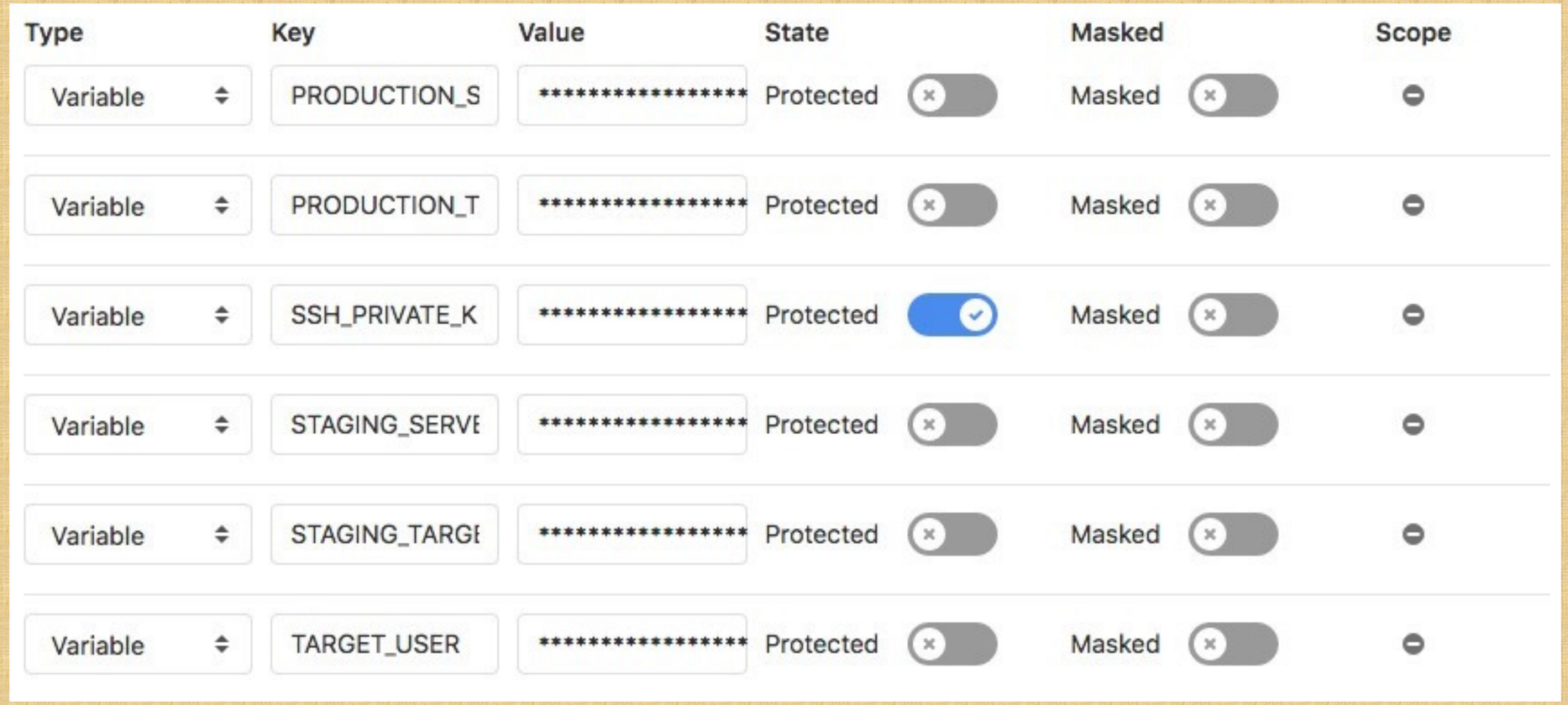
Highly Configurable

docs.gitlab.com/ee/ci/yaml

- Specify a base Docker image Determine conditions
- under which jobs should run Define dependencies
- between jobs Pre/post-script hooks Dynamic
- environments Automatically tag new releases
-
-

Author: Nho Luong
Skill: DevOps Engineer Lead

Environment Variables



Author: Nho Luong
Skill: DevOps Engineer Lead

Simple Deployments

Author: Nho Luong
Skill: DevOps Engineer Lead

What Does our Deployment Process Look Like Right Now?

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

Author: Nho Luong
Skill: DevOps Engineer Lead

A Bare-Bones Deployment Method

1.Build the app

2 scp a tarball to production

3.rsync the files into the web root

Author: Nho Luong
Skill: DevOps Engineer Lead

Archive, scp, and rsync

```
ship_to_production:

    stage: deploy
    script:
        - tar -czf release.tgz dist/* - scp release.tgz
        "${SSH_USER}@${PRODUCTION_SERVER}:/tmp" - ssh
        "${SSH_USER}@${PRODUCTION_SERVER}" "cd /tmp \
          && tar -xzf release.tgz \
          && rsync release/ /var/www/html/"
```

Author: Nho Luong
Skill: DevOps Engineer Lead

Atomic Deployments

releases/

Contains multiple, timestamped deployments

shared/

Data that should persist between deployments

current

Symlink to the current release in releases/, part of web root

Author: Nho Luong
Skill: DevOps Engineer Lead

Atomic Deployments

/var/www releases/

+ 1563759801/
# Oldest + 1563759802/ + 1563759803/
# Newest <= current symlink

shared/
+ logs/ +
uploads/
- .env

Author: Nho Luong
Skill: DevOps Engineer Lead

Atomic Deployments

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

```
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
```

Author: Nho Luong
Skill: DevOps Engineer Lead

Extending Our Pipelines

Author: Nho Luong
Skill: DevOps Engineer Lead

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

Author: Nho Luong
Skill: DevOps Engineer Lead

Automate Release Chores

Make your pipeline work for you!

- Automatically build Docker images for your app
- Generate + publish
- API documentation
- Code coverage reporting Webhooks
- If you can script it, you can do it!

Author: Nho Luong
Skill: DevOps Engineer Lead

Remember: This Isn't Magic!

Author: Nho Luong
Skill: DevOps Engineer Lead

Thank You

Author: Nho Luong
Skill: DevOps Engineer Lead