CI/CD Pipeline mit Gitlab CI und Kubernetes

Johannes M. Scheuermann
Johannes M. Scheuermann
Cloud Platform Engineer @ inovex

› Software-Defined Datacenters
› Infrastructure Services
› High Availability /Scalability / Resilience
› Cloud technologies
What about you?
Motivation
Scenario: ToDo Webapp
Introduction

- Use Case: Managing Tasks
- HTML/CSS/JS, Golang
- Redis (Master/Slave)
Technologies

Gitlab

Docker

Kubernetes
How to build your binary?
Container-based Build Slaves
Gitlab CI
Pipeline

Push Code ➔ Build Artifact ➔ Trigger Tests ➔ Release Artifact ➔ Deploy
Pipeline

Commit

Tag
Pipeline
Configuration

```yaml
build-app:go1.7:
  stage: build
  image: golang:1.7
  variables:
    go_version: "1.7"
  <<: *build_app_yaml_definition

.build_app_yaml_template: &build_app_yaml_definition # Hidden job that defines
  tags:
    - shared
  artifacts:
    expire_in: 2 days
    paths:
      - bin/*.bin
  script:
    - export GOPATH="$HOME/go"
    - export GOBIN="$GOPATH/bin"
    - mkdir -p $GOPATH $GOBIN
    - go env
    - go get
    - artifact_version=${CI_BUILD_TAG}
    - test -z "${artifact_version}" && artifact_version=${CI_BUILD_REF:0:7}
    - CGO_ENABLED=0 GOOS=linux GOARCH=amd64 go build -ldflags "-s -w -X main.a"
```
Pipeline
Configuration

```yaml
release-app:
  stage: release
  image: docker:latest #TODO latest?
  tags:
    - shared
  only:
    - tags
dependencies: # Retrieve Artifacts (http://docs.gitlab.com/ce/ci/yaml/README.html#dependencies)
  - build-app:go1.7
variables:
  go_version: "1.7"
script:
  - ln -s todo-app.go${go_version?}.bin bin/todo-app
  - docker login -u _json_key -p "${REGISTRY_JSON_KEYFILE?}" https://eu.gcr.io
  - image_name="eu.gcr.io/${CI_KBS_PROJECT?}/todo-app-web:${CI_BUILD_TAG?}"
  - docker build -t ${image_name?} .
  - docker push ${image_name?}
  - echo "Your new image is '${image_name?}'. Have fun!"
```
Typical CI/CD Problems

- Build-Software dependencies
- Reproducible builds
- Automated Rollout
- Rollbacks
- Isolated feature testing
Deploy to Kubernetes
Kubernetes Deployment

Deploy v1.2 into staging

Kubernetes

ns: todo-app-staging
Pod v1.1
Pod v1.1

ns: todo-app-production
Pod v1.1
Pod v1.1
Kubernetes Deployment

Deploy v1.2 into staging

API

Gitlab CI

Kubernetes

ns: todo-app-staging

Pod v1.1

Pod v1.1

Pod v1.2

ns: todo-app-production

Pod v1.1

Pod v1.1
Kubernetes Deployment

Deploy v1.2 into staging

Gitlab CI

Kubernetes

ns: todo-app-staging

Pod v1.1

Pod v1.2

ns: todo-app-production

Pod v1.1

Pod v1.1
Kubernetes Deployment

Deploy v1.2 into staging

ns: todo-app-staging
Pod v1.1
Pod v1.2

ns: todo-app-production
Pod v1.1
Pod v1.1
Kubernetes Deployment

Deploy v1.2 into staging

Gitlab CI → Kubernetes

**ns: todo-app-staging**

Pod v1.2

Pod v1.2

**ns: todo-app-production**

Pod v1.1

Pod v1.1
Pipeline
Configuration

```yaml
# DEPLOY

deploy-into-staging:
  stage: deploy-staging
  environment: staging
  image: docker:latest #TODO latest?
  tags:
    - shared
  only:
    - tags
  when: manual
variables:
  k8s_namespace: todo-app-staging
<<: *get_kubectl_yaml_template #inserts before_script definition
script:
  - image_name="eu.gcr.io/${CI_K8S_PROJECT?}/todo-app-web:${CI_BUILD_TAG?}"
  - kubectl config use-context default-context-staging
  - kubectl --namespace=${k8s_namespace?} set image deployment/todo-app todo-app=${image_name?}

Typical CI/CD Problems

- Build-Software dependencies
- Reproducible builds
- Automated Rollout
- Rollbacks
- Isolated feature testing
## Environments

<table>
<thead>
<tr>
<th>Environment</th>
<th>Last deployment</th>
<th>Job</th>
<th>Commit</th>
<th>Updated</th>
<th>Action</th>
</tr>
</thead>
</table>
| production  | #17 by 🦊     | deploy-into-production #51 49596 | 🟢 1d8d17e5  
Revert "updating html template" | 4 months ago | Re-deploy |
| staging     | #15 by 🦊      | deploy-into-staging #5149 518 | 🟢 f0ff04cf  
make fancy title | 4 months ago | Re-deploy |
## Rollback

<table>
<thead>
<tr>
<th>ID</th>
<th>Commit</th>
<th>Job</th>
<th>Created</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>#17</td>
<td><code>v0.3 → 1d8d17e5</code></td>
<td>Revert &quot;updating html templat...&quot;</td>
<td>4 months ago</td>
<td>Re-deploy</td>
</tr>
<tr>
<td>#16</td>
<td><code>v0.5 → fdf04cf</code></td>
<td>deploy-into-production (#5149519)</td>
<td>4 months ago</td>
<td>Rollback</td>
</tr>
<tr>
<td>#14</td>
<td><code>v0.3 → 1d8d17e5</code></td>
<td>Revert &quot;updating html templat...&quot;</td>
<td>4 months ago</td>
<td>Rollback</td>
</tr>
</tbody>
</table>
Typical CI/CD Problems

★ Build-Software dependencies ✔
★ Reproducible builds ✔
★ Automated Rollout ✔
★ Rollbacks ✔
★ Isolated feature testing
## Review Environments

<table>
<thead>
<tr>
<th>Environment</th>
<th>Last deployment</th>
<th>Job</th>
<th>Commit</th>
<th>Updated</th>
<th>Re-deploy</th>
</tr>
</thead>
</table>
| production          | #17 by 🌀        | deploy-into-production #51 49596 | ▶️ 1d8d17e5  
Revert "updating html template" | 4 months ago |          |
| review/feature-4711 | #22 by 🌀        | deploy-into-review #12098 976 | ▶️ 3fc1bad9  
Update .gitlab-ci.yml | 6 minutes ago | Re-deploy |
| review/feature-1337 | #21 by 🌀        | deploy-into-review #12098 959 | ▶️ 3fc1bad9  
Update .gitlab-ci.yml | 6 minutes ago | Re-deploy |
| staging             | #15 by 🌀        | deploy-into-staging #5149 518 | ▶️ fdff04cf  
makes fancy title | 4 months ago   |          |
deploy-into-review:
  stage: deploy-review
  environment: review/$CI_BUILD_REF_NAME
  image: docker:latest #TODO latest?
  tags:
    - shared
  only:
    - branches
  except:
    - master
  script:
    - echo "deploy ${CI_BUILD_REF_NAME} into environment."
Typical CI/CD Problems

- Build-Software dependencies
- Reproducible builds
- Automated Rollout
- Rollbacks
- Isolated feature testing
Demo Time: Video!

youtu.be/Ly_f221O6Fo
Assets & Links

gitlab.com/inovex/pa-todo-app-web