

HashiConf Europe

8-11 June 2021 (CEST)

FEATURED SPEAKER

Tracey Jaquith

Internet Archive

GitLab + Nomad = A Dream Come True





GitLab + Nomad = A Dream Come True

HashiConf Europe 2021
Tracey Jaquith

slides: tracey.dev.archive.org

@tracey_pooh 🐻
Internet Archive 🏛



@tracey_pooh 🐻
Internet Archive 🏛

TV Architect
· video/audio · devops · javascript ·



@tracey_pooh 🐻
Internet Archive 🏛

TV Architect
· video/audio · devops · javascript ·

slides: <https://tracey.dev.archive.org>
repo: gitlab/internetarchive

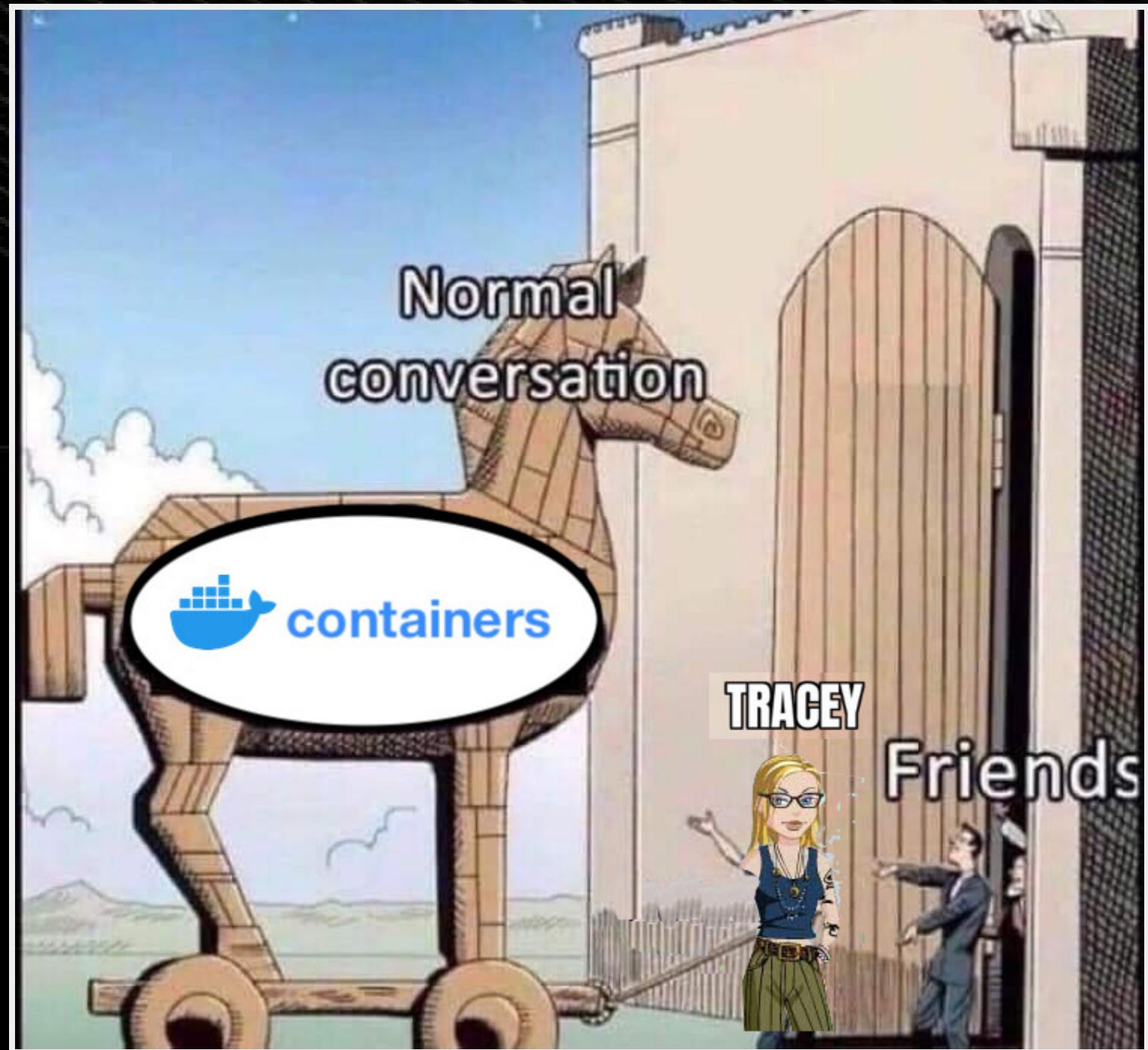


@tracey_pooh 🐻
Internet Archive 🏛

TV Architect
· video/audio · devops · javascript ·

slides: <https://tracey.dev.archive.org>
repo: gitlab/internetarchive





tracey.dev.archive.org



WHY ARE WE HERE?



WHY ARE WE HERE?

Use GitLab with Nomad clusters



GITLAB + NOMAD



GITLAB + NOMAD

2 line .gitlab-ci.yml



GITLAB + NOMAD

2 line .gitlab-ci.yml

2 CI/CD variables



APPROACH & GOALS



APPROACH & GOALS

nomad replace kubernetes



APPROACH & GOALS

nomad replace kubernetes

Use GitLab [build] phase
(build & docker push to registry)



APPROACH & GOALS

nomad replace kubernetes

Use GitLab [build] phase
(build & docker push to registry)

Custom [deploy] phase



APPROACH & GOALS

nomad replace kubernetes

Use GitLab [build] phase
(build & docker push to registry)

Custom [deploy] phase



APPROACH & GOALS

nomad replace kubernetes

Use GitLab [build] phase
(build & docker push to registry)

Custom [deploy] phase

tracey.dev.archive.org

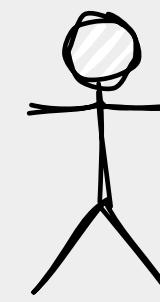


How we'll do it



commit & push

How we'll do it



commit & push
creates GitLab CI/CD pipeline

How we'll do it



commit & push
creates GitLab CI/CD pipeline
[build] phase

How we'll do it



commit & push
creates GitLab CI/CD pipeline
[test] phase

How we'll do it



commit & push
creates GitLab CI/CD pipeline
[**deploy**] phase

How we'll do it



commit & push
creates GitLab CI/CD pipeline
[**deploy**] phase

GitLab CI/CD vars

`$CI_*`

Custom env vars

`$NOMAD_VAR_*`

How we'll do it



commit & push
creates GitLab CI/CD pipeline
[**deploy**] phase

GitLab CI/CD vars

eg:

\$CI_COMMIT_SHA

Custom env vars

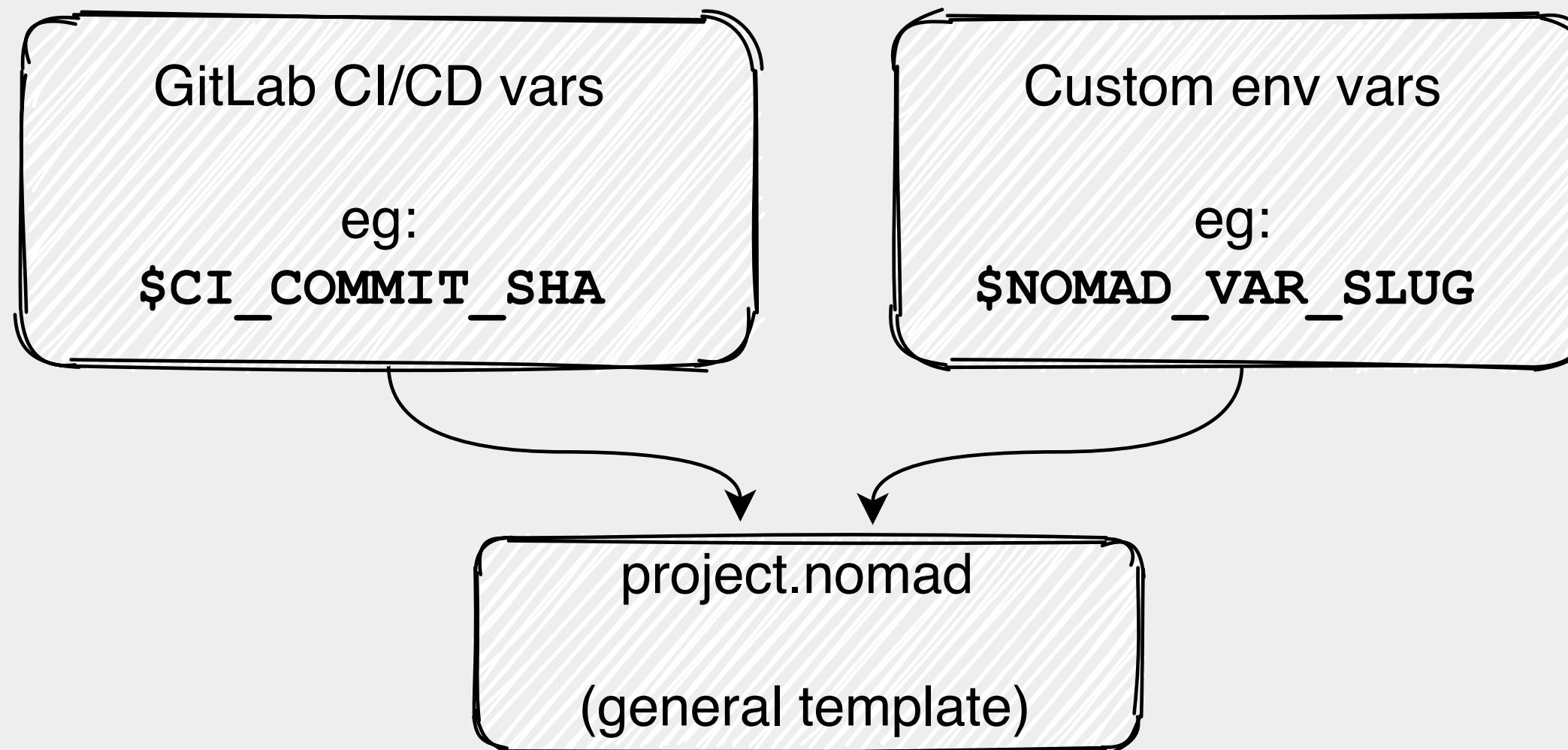
eg:

\$NOMAD_VAR_SLUG

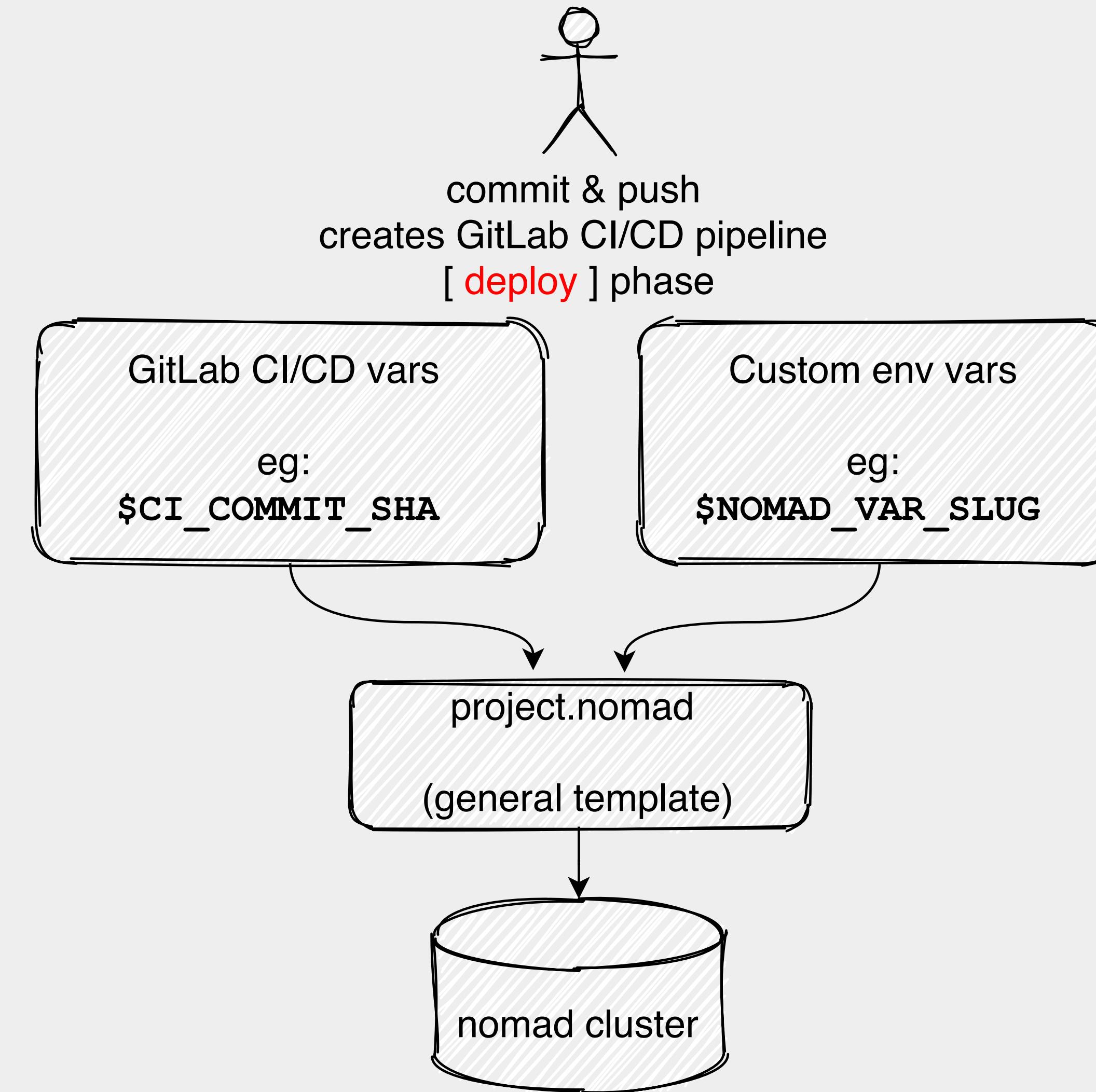
How we'll do it



commit & push
creates GitLab CI/CD pipeline
[**deploy**] phase



How we'll do it



TL;DR 😵

GitLab repository's
⚙️ Settings > CI/CD > Variables

NOMAD_ADDR <https://pod.archive.org:4646>

NOMAD_TOKEN 133t-z6rf-1234-1243bcd35199187



TL;DR 😊

```
include:  
- remote: 'https://gitlab.com/internetarchive/nomad/-/raw/master/.gitlab-ci.yml'
```

add 👉 to your GitLab repository :

- top-level file:
- .gitlab-ci.yml



THE INCLUDED .GITLAB-CI.YML



THE INCLUDED .GITLAB-CI.YML

```
deploy:  
  stage: deploy  
  script:
```



THE INCLUDED .GITLAB-CI.YML

```
deploy:  
  stage: deploy  
  script:  
  
    # ---- get templated project.nomad ----  
    - wget gitlab.com/internetarchive/nomad/-/raw/master/project.nomad  
    - mv project.nomad project.hcl  
    - sed -i "s/NOMAD_VAR_SLUG/$NOMAD_VAR_SLUG/" project.hcl
```



THE INCLUDED .GITLAB-CI.YML

```
deploy:  
  stage: deploy  
  script:  
  
    # ---- get templated project.nomad ----  
    - wget gitlab.com/internetarchive/nomad/-/raw/master/project.nomad  
    - mv project.nomad project.hcl  
    - sed -i "s/NOMAD_VAR_SLUG/$NOMAD_VAR_SLUG/" project.hcl  
  
    # ---- write GitLab env vars starting with "CI_" to env var file ----  
    - node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("CI_")))))' >| /tmp/env.json
```



THE INCLUDED .GITLAB-CI.YML

```
deploy:
  stage: deploy
  script:

    # ---- get templated project.nomad ----
    - wget gitlab.com/internetarchive/nomad/-/raw/master/project.nomad
    - mv project.nomad project.hcl
    - sed -i "s/NOMAD_VAR_SLUG/$NOMAD_VAR_SLUG/" project.hcl

    # ---- write GitLab env vars starting with "CI_" to env var file ----
    - node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("CI_")))))' >| /tmp/env.json

    # ---- write env vars starting w/ "NOMAD_SECRET_" to env var file ----
    - echo NOMAD_SECRETS=$(node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("NOMAD_SECRET_"))).map(
```



THE INCLUDED .GITLAB-CI.YML

```
deploy:
  stage: deploy
  script:

    # ---- get templated project.nomad ----
    - wget gitlab.com/internetarchive/nomad/-/raw/master/project.nomad
    - mv project.nomad project.hcl
    - sed -i "s/NOMAD_VAR_SLUG/$NOMAD_VAR_SLUG/" project.hcl

    # ---- write GitLab env vars starting with "CI_" to env var file ----
    - node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("CI_")))))' >| /tmp/env.json

    # ---- write env vars starting w/ "NOMAD_SECRET_" to env var file ----
    - echo NOMAD_SECRETS=$(node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("NOMAD_SECRET_"))).map(
```

```
# ---- deploy job to nomad ----
- nomad run -var-file=/tmp/env.json -var-file=/tmp/env.env project.hcl
```



THE INCLUDED .GITLAB-CI.YML

```
deploy:
  stage: deploy
  script:

    # ---- get templated project.nomad ----
    - wget gitlab.com/internetarchive/nomad/-/raw/master/project.nomad
    - mv project.nomad project.hcl
    - sed -i "s/NOMAD_VAR_SLUG/$NOMAD_VAR_SLUG/" project.hcl

    # ---- write GitLab env vars starting with "CI_" to env var file ----
    - node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("CI_")))))' >| /tmp/env.json

    # ---- write env vars starting w/ "NOMAD_SECRET_" to env var file ----
    - echo NOMAD_SECRETS=$(node -e 'console.log(JSON.stringify(Object.fromEntries(Object.entries(process.env).filter(([k, v]) => k.startsWith("NOMAD_SECRET_"))).map(k))')) >| /tmp/env.secret

    # ---- deploy job to nomad ----
    - nomad run -var-file=/tmp/env.json -var-file=/tmp/env.env project.hcl
```

see full .gitlab-ci.yml for more



NOMAD PROJECT

- generic template: `project.nomad`



NOMAD PROJECT

- generic template: `project.nomad`
- uses template vars + GitLab CI/CD Auto DevOps variables



NOMAD PROJECT

- generic template: `project.nomad`
- uses template vars + GitLab CI/CD Auto DevOps variables

```
$CI_COMMIT_REF_SLUG  
$CI_REGISTRY_IMAGE  
..  
$NOMAD_VAR_MEMORY  
$NOMAD_SECRET_DATABASE_URL
```



NOMAD PROJECT

- generic template: `project.nomad`
- uses template vars + GitLab CI/CD Auto DevOps variables

```
$CI_COMMIT_REF_SLUG  
$CI_REGISTRY_IMAGE  
..  
$NOMAD_VAR_MEMORY  
$NOMAD_SECRET_DATABASE_URL
```

- leverages: `HCL v2`



NOMAD PROJECT

- generic template: `project.nomad`
- uses template vars + GitLab CI/CD Auto DevOps variables

```
$CI_COMMIT_REF_SLUG  
$CI_REGISTRY_IMAGE  
..  
$NOMAD_VAR_MEMORY  
$NOMAD_SECRET_DATABASE_URL
```

- leverages: `HCL v2`

```
network {  
    dynamic "port" {  
        # port.key == portnumber  port.value == portname  
        for_each = merge(var.PORTS, var.PG, {})  
        labels = [ "${port.value}" ]  
        content {  
            to = port.key  
        }  
    }  
}
```



MINIMAL PROJECT USABLE WITH GITLAB



```
job "hello-world" {  
    datacenters = ["dc1"]  
    group "hello-world" {  
        network { port "http" { to = 5000 } }  
    }
```



```
job "hello-world" {
  datacenters = ["dc1"]
  group "hello-world" {
    network { port "http" { to = 5000 } }
  }
}

task "hello-world" {
  driver = "docker"
  config {
    image = "${var.CI_REGISTRY_IMAGE}/${var.CI_COMMIT_REF_SLUG}:${var.CI_COMMIT_SHA}"
    ports = [ "http" ]
  }
}
```



```
job "hello-world" {
  datacenters = ["dc1"]
  group "hello-world" {
    network { port "http" { to = 5000 } }
  }
}

task "hello-world" {
  driver = "docker"
  config {
    image = "${var.CI_REGISTRY_IMAGE}/${var.CI_COMMIT_REF_SLUG}:${var.CI_COMMIT_SHA}"
    ports = [ "http" ]
  }
}

auth {
  server_address = "${var.CI_REGISTRY}"
  username = "${var.CI_REGISTRY_USER}"
  password = "${var.CI_REGISTRY_PASSWORD}"
}
}
```



```
job "hello-world" {
  datacenters = ["dc1"]
  group "hello-world" {
    network { port "http" { to = 5000 } }

  task "hello-world" {
    driver = "docker"
    config {
      image = "${var.CI_REGISTRY_IMAGE}/${var.CI_COMMIT_REF_SLUG}:${var.CI_COMMIT_SHA}"
      ports = [ "http" ]

      auth {
        server_address = "${var.CI_REGISTRY}"
        username = "${var.CI_REGISTRY_USER}"
        password = "${var.CI_REGISTRY_PASSWORD}"
      }}
    }

    service {
      name = "hello-world"
      tags = [ "urlprefix-${var.CI_PROJECT_PATH_SLUG}-${var.CI_COMMIT_REF_SLUG}.${var.KUBE_INGRESS_BASE_DOMAIN}:443/" ]
      port = "http"
    }
  }
}
```



```
job "hello-world" {
  datacenters = ["dc1"]
  group "hello-world" {
    network { port "http" { to = 5000 } }

    task "hello-world" {
      driver = "docker"
      config {
        image = "${var.CI_REGISTRY_IMAGE}/${var.CI_COMMIT_REF_SLUG}:${var.CI_COMMIT_SHA}"
        ports = [ "http" ]

        auth {
          server_address = "${var.CI_REGISTRY}"
          username = "${var.CI_REGISTRY_USER}"
          password = "${var.CI_REGISTRY_PASSWORD}"
        }}
      }

      service {
        name = "hello-world"
        tags = [ "urlprefix-${var.CI_PROJECT_PATH_SLUG}-${var.CI_COMMIT_REF_SLUG}.${var.KUBE_INGRESS_BASE_DOMAIN}:443/" ]
        port = "http"
      }

      check {
        type      = "http"
        port      = "http"
        path      = "/"
        interval  = "10s"
        timeout   = "2s"
      }}}}
```



HAS NOMAD CLUSTER? TRY THIS NOW! 😊

```
NOMAD_ADDR=...  
NOMAD_TOKEN=...
```

```
wget https://gitlab.com/internetarchive/nomad/-/raw/master/etc/hello-world.hcl  
nomad run hello-world.hcl
```



NOMAD_VAR_SLUG 

Semantic hostnames 

`https://services-timemachine`
 └─── group └─── project



NOMAD_VAR_SLUG

Semantic hostnames 

`https://services-timemachine`
 └───
 |
 | (GitLab) group
 | project



NOMAD_VAR_SLUG 🐌

Semantic hostnames 🙌

`https://services-timemachine`
 └───┐
 └───┘ (GitLab) group project

`https://frontend-website-123-ticket-3939-fix-boom`
 └───┐
 └───┘ group project branch



NOMAD JOB NAMES

GET SAME SEMANTIC SLUG

```
$ nomad status
```

ID	Type	Priority	Status	Submit Date
fabio	system	50	running	2020-07-04T12:14:32-07:00
internetarchive-word-salad	service	50	running	2020-07-04T04:24:31-07:00
internetarchive-www	service	50	running	2020-07-04T04:36:02-07:00
internetarchive-www-mobile-nav	service	50	running	2020-07-04T13:59:16-07:00
services-timemachine	service	50	running	2020-07-04T14:00:05-07:00
x-thumb	service	50	running	2020-07-04T04:34:54-07:00



NOMAD JOB NAMES

GET SAME SEMANTIC SLUG

```
$ nomad status
```

ID	Type	Priority	Status	Submit Date
fabio	system	50	running	2020-07-04T12:14:32-07:00
internetarchive-word-salad	service	50	running	2020-07-04T04:24:31-07:00
internetarchive-www	service	50	running	2020-07-04T04:36:02-07:00
internetarchive-www-mobile-nav	service	50	running	2020-07-04T13:59:16-07:00
services-timemachine	service	50	running	2020-07-04T14:00:05-07:00
x-thumb	service	50	running	2020-07-04T04:34:54-07:00

job name



DEMO TIME



- Create new gitlab repo
- Deploy to nomad



HELLO MY WORLD

3 FILES, 10 LINES



HELLO MY WORLD

3 FILES, 10 LINES

```
└── clock.js └──  
var http = require('http')  
http.createServer((req, res) => {  
  res.end(`the time is: ${new Date()}`)  
}).listen(5000)
```



HELLO MY WORLD

3 FILES, 10 LINES

```
└── clock.js └──  
var http = require('http')  
http.createServer((req, res) => {  
  res.end(`the time is: ${new Date()}`)  
}).listen(5000)
```

```
└── Dockerfile └──  
FROM node:alpine  
COPY . /app  
CMD node /app/clock.js
```



HELLO MY WORLD

3 FILES, 10 LINES

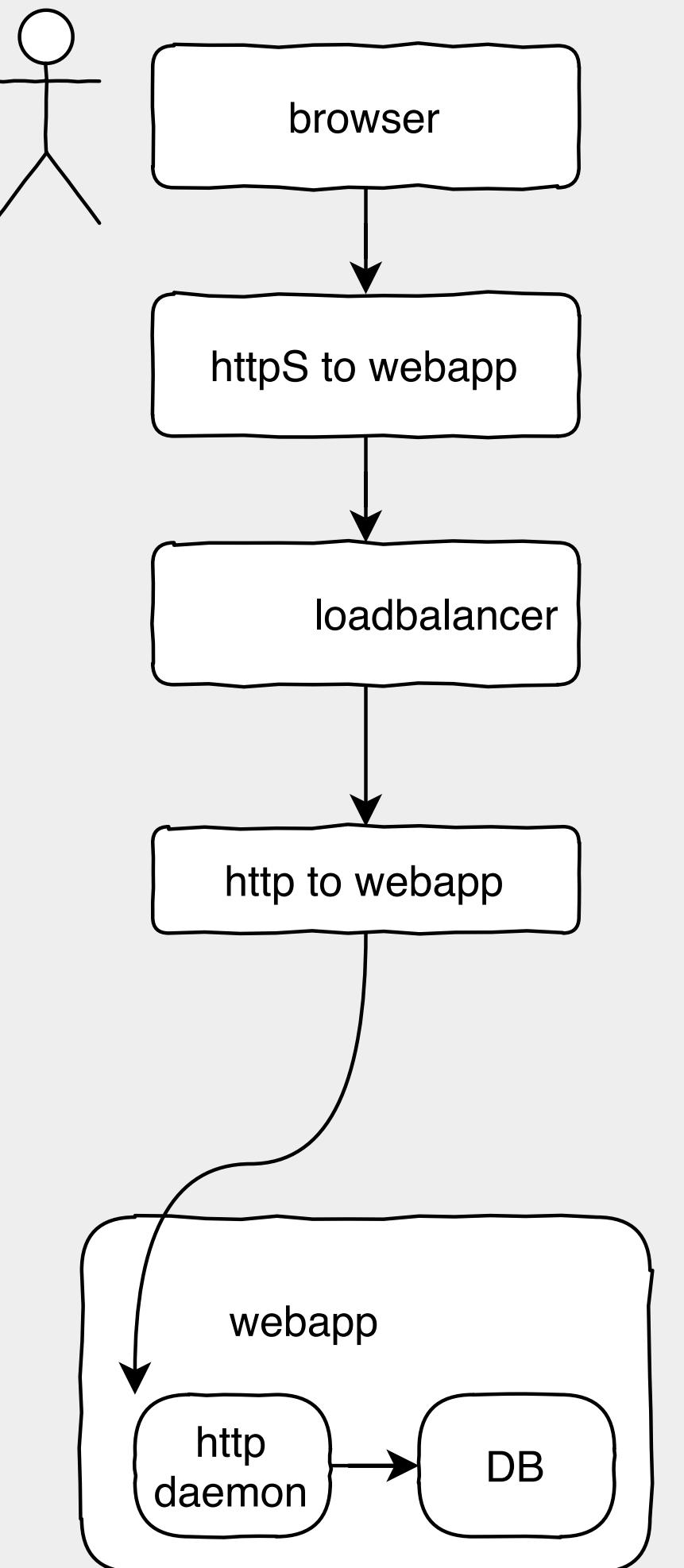
```
└── clock.js └──  
var http = require('http')  
http.createServer((req, res) => {  
  res.end(`the time is: ${new Date()}`)  
}).listen(5000)
```

```
└── Dockerfile └──  
FROM node:alpine  
COPY . /app  
CMD node /app/clock.js
```

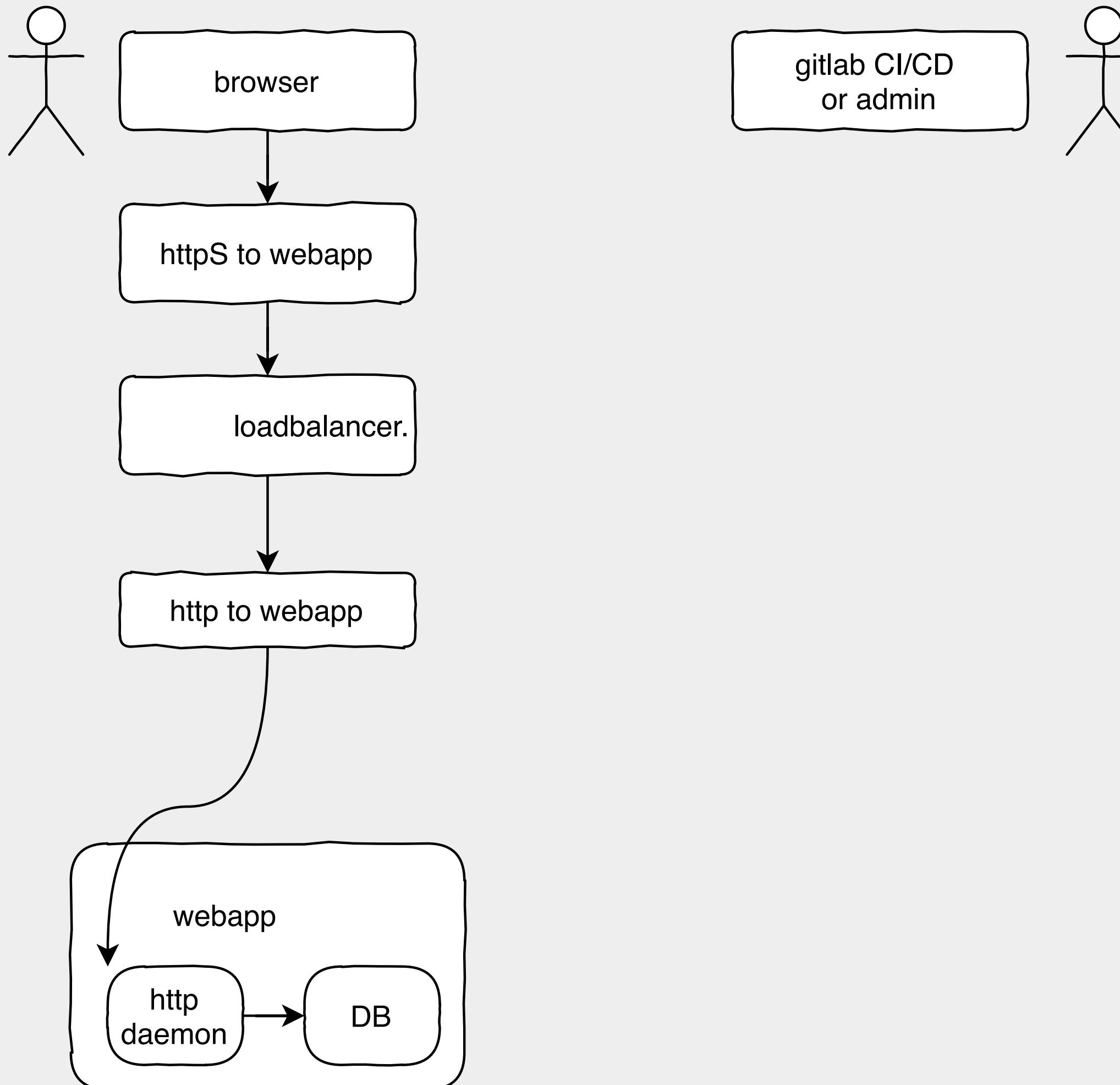
```
└── .gitlab-ci.yml └──  
include:  
- remote: 'https://gitlab.com/internetarchive/nomad/-/raw/master/.gitlab-ci.yml'
```



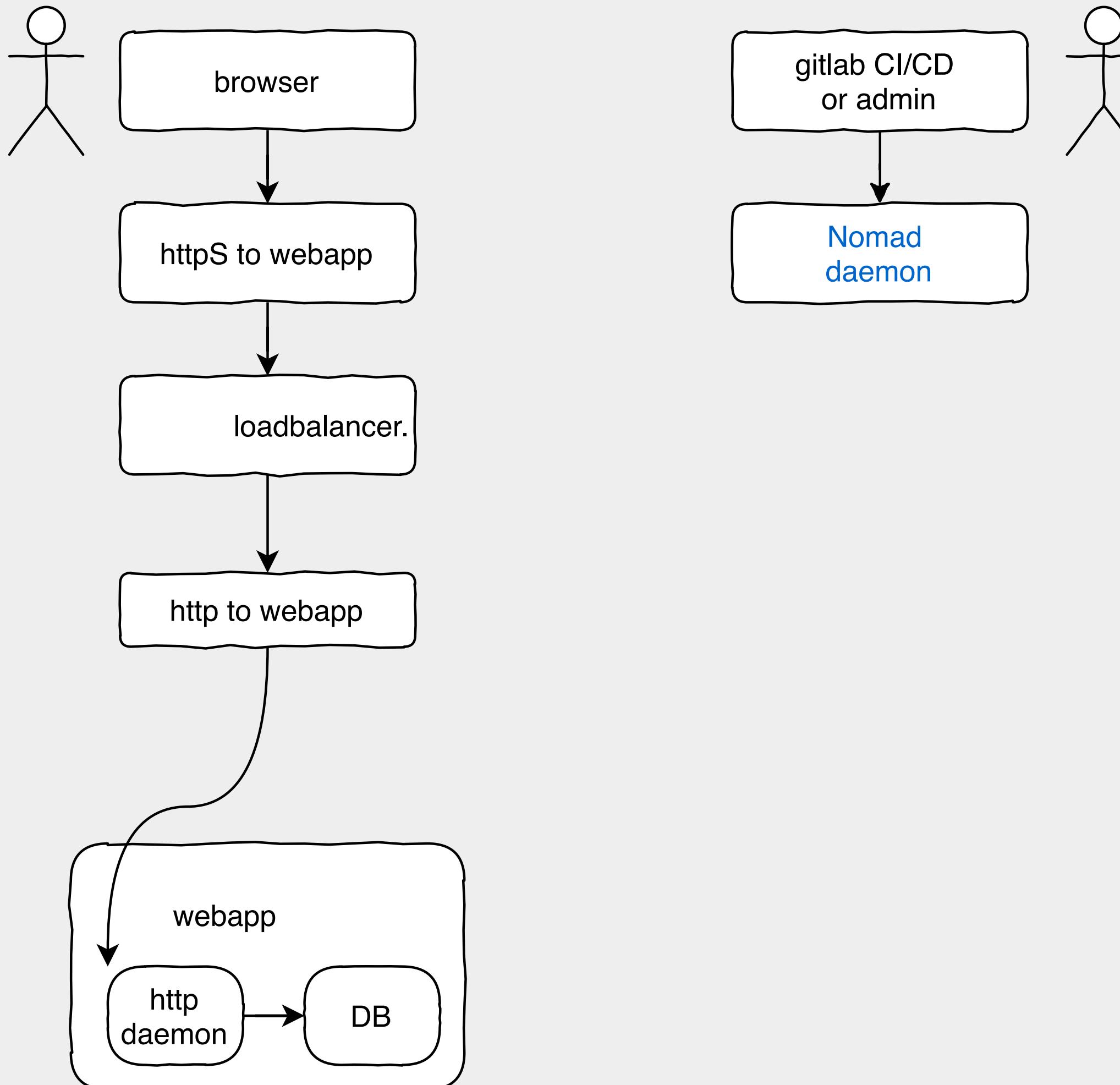
Hashistack



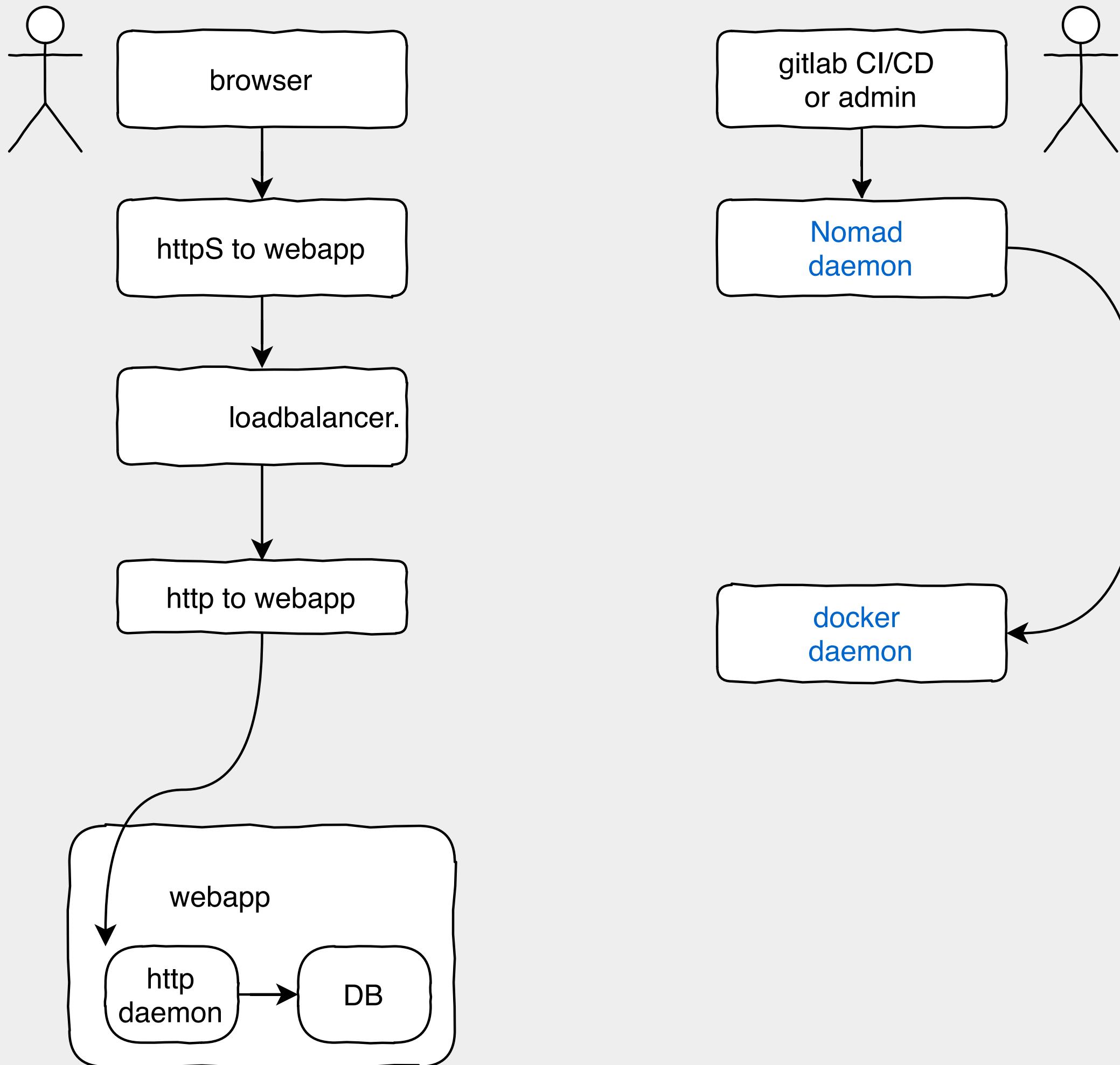
Hashistack



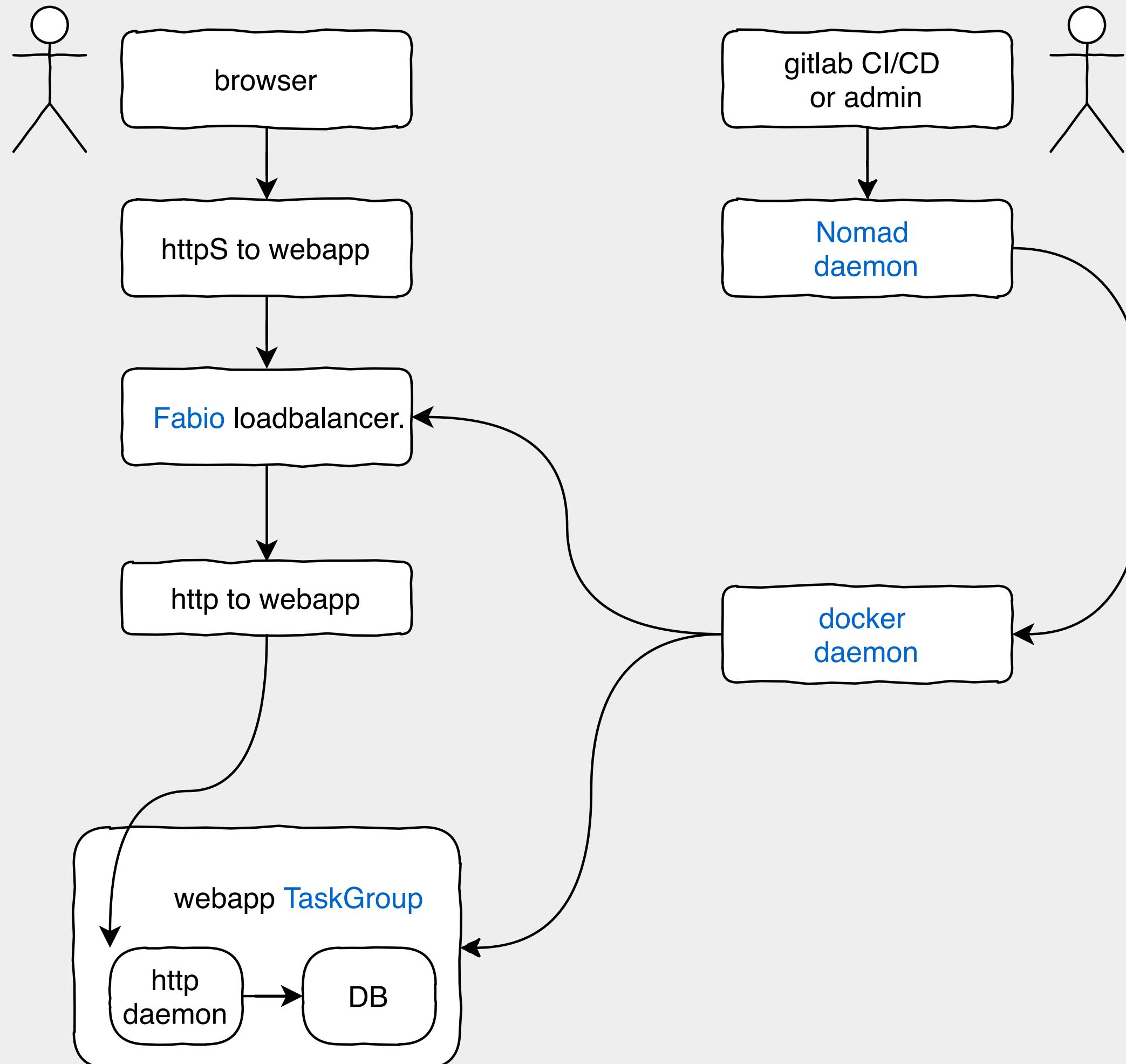
Hashistack



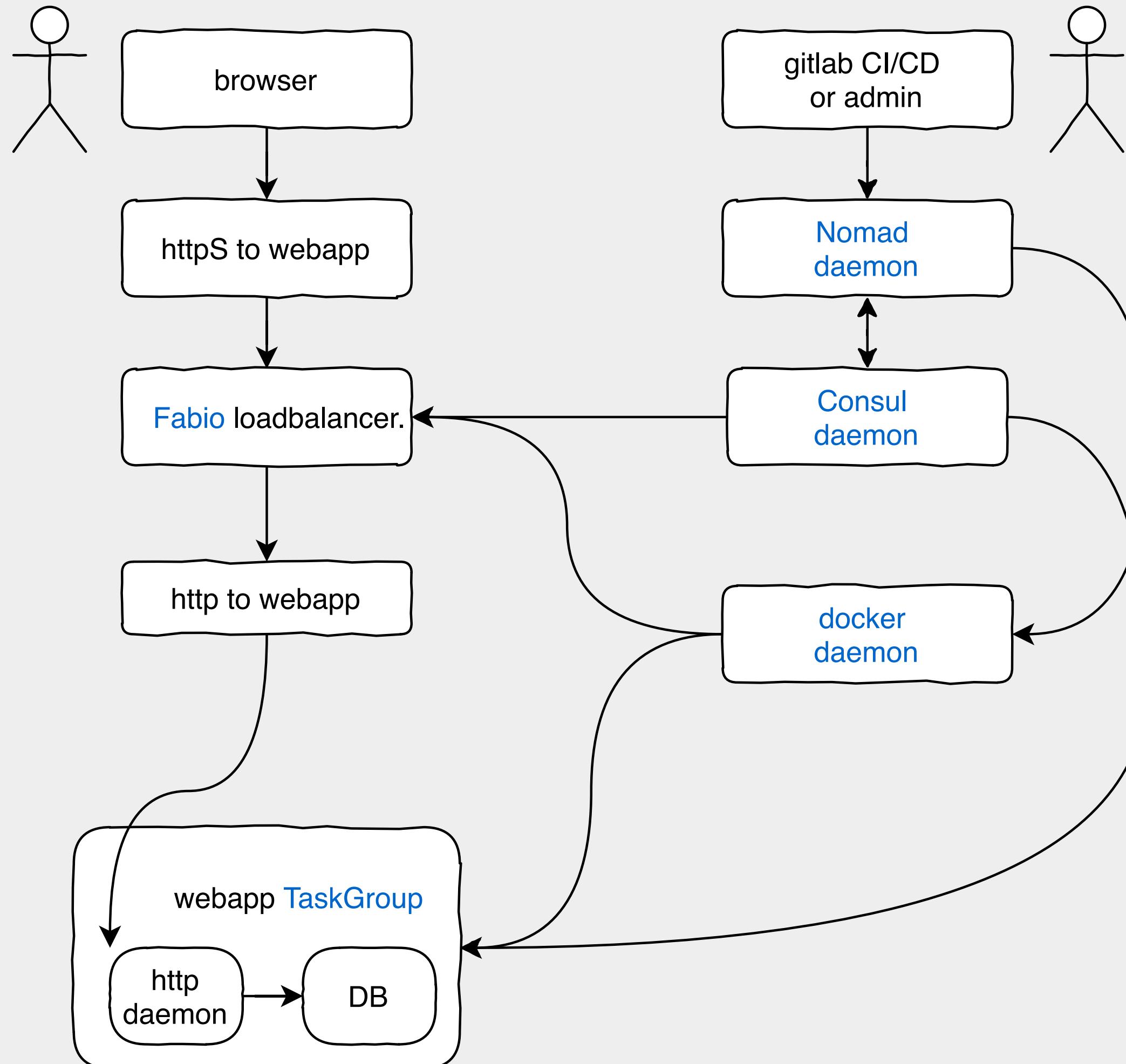
Hashistack



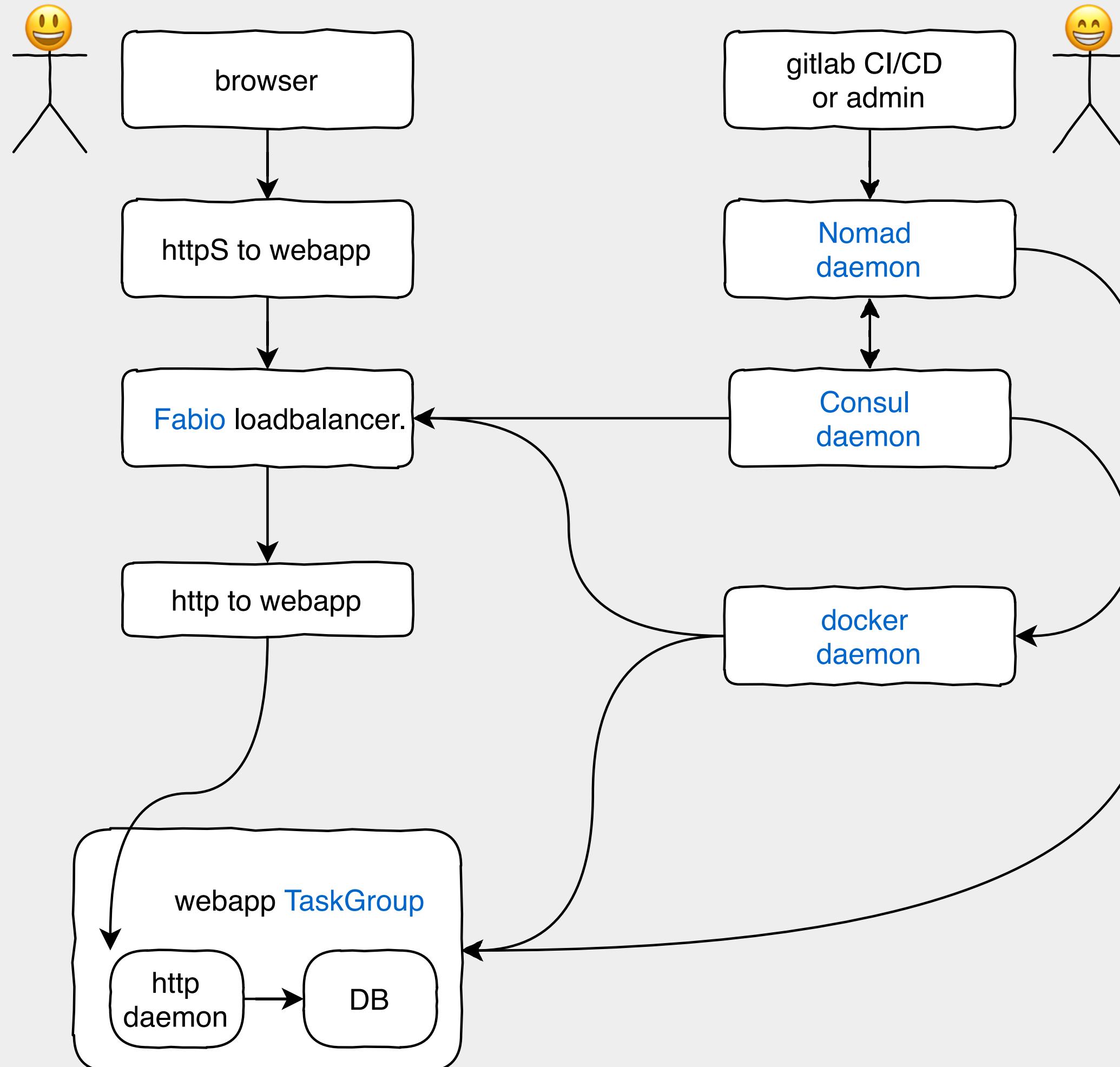
Hashistack



Hashistack



Hashistack



<https://gitlab.com/internetarchive/hashiconf-demo>



FULL PROJECT.NOMAD

<https://gitlab.com/internetarchive/nomad/project.nomad>



CUSTOMIZE

Optional CI/CD vars for **project.nomad**

```
NOMAD_VAR_BIND_MOUNTS  
NOMAD_VAR_CHECK_PROTOCOL  
NOMAD_VAR_COUNT  
NOMAD_VAR_CPU  
NOMAD_VAR_HEALTH_TIMEOUT  
NOMAD_VAR_HOME  
NOMAD_VAR_HOSTNAMES  
NOMAD_VAR_MEMORY  
NOMAD_VAR_NO_DEPLOY  
NOMAD_VAR_PG  
NOMAD_VAR_PORTS  
NOMAD_VAR_PV  
NOMAD_VAR_PV_DB
```



CUSTOMIZE

Optional CI/CD vars for [project.nomad](#)

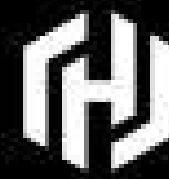
```
NOMAD_VAR_BIND_MOUNTS  
NOMAD_VAR_CHECK_PROTOCOL  
NOMAD_VAR_COUNT  
NOMAD_VAR_CPU  
NOMAD_VAR_HEALTH_TIMEOUT  
NOMAD_VAR_HOME  
NOMAD_VAR_HOSTNAMES  
NOMAD_VAR_MEMORY  
NOMAD_VAR_NO_DEPLOY  
NOMAD_VAR_PG  
NOMAD_VAR_PORTS  
NOMAD_VAR_PV  
NOMAD_VAR_PV_DB
```

Add any  to your [.gitlab-ci.yml](#)
[\(more info\)](#)



ISSUES WITH: docker login?

- Create group `read_registry` Deploy Token



ISSUES WITH: docker login?

- Create group `read_registry` Deploy Token
 - Add to Group:  Settings > CI/CD > Variables
 - `CI_R2_USER`
 - `CI_R2_PASS`



MONITORING & USAGE



aliases based on CWD & git info

nom-ssh ssh into container

nom-cp "hot copy" file laptop → container

nom-app open webapp url

nom-status deploy info



HALP 😟 - [BUILD] CACHING

- docker.sock not docker-in-docker
- gitlab runners & monorepos



ADDITIONAL HELP

Create Nomad cluster
linux, digital ocean (\$5), mac/laptop



ADDITIONAL HELP

Create Nomad cluster
linux, digital ocean (\$5), mac/laptop
create domain https certs



ADDITIONAL HELP

[Create Nomad cluster](#)
linux, digital ocean (\$5), mac/laptop

[create domain https certs](#)
[gitlab pipeline examples](#)



RESULTS:  MOVE TO NOMAD



RESULTS:  MOVE TO NOMAD

70 repos/webapps deployed



RESULTS: MOVE TO NOMAD

70 repos/webapps deployed

400+ review apps deployed



RESULTS: MOVE TO NOMAD

70 repos/webapps deployed

400+ review apps deployed

2x+ faster deploys



RESULTS: MOVE TO NOMAD

70 repos/webapps deployed

400+ review apps deployed

2x+ faster deploys

all main dev using gitlab + nomad review apps



DEPLOY WITH GITLAB + NOMAD

- 2 line `.gitlab-ci.yml`
- 2 CI/CD vars



DEPLOY WITH GITLAB + NOMAD

- 2 line `.gitlab-ci.yml`
- 2 CI/CD vars





THE END



Thank You!
slides: tracey.dev.archive.org
repo: [gitlab/internetarchive](https://gitlab.internetarchive.org/internetarchive)