



Максим Залысин

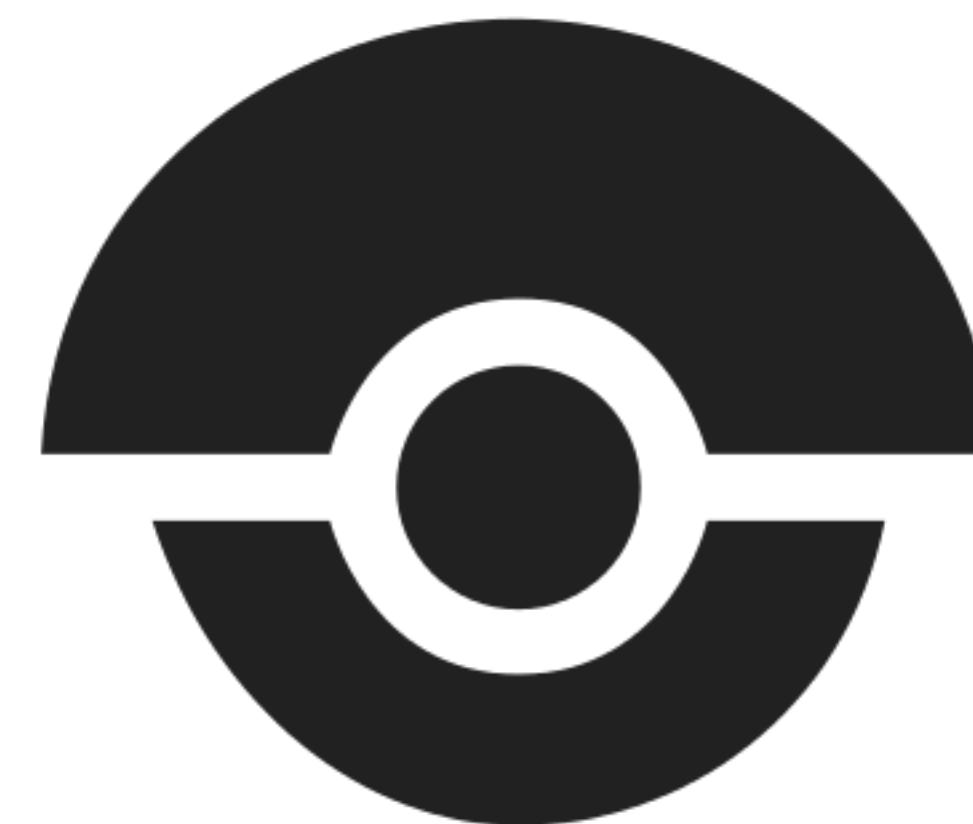
- ```
cli: 🍏 🍷 🍹
version: v4.2.1
```



integrity



# Jenkins



GitLab



# Travis CI





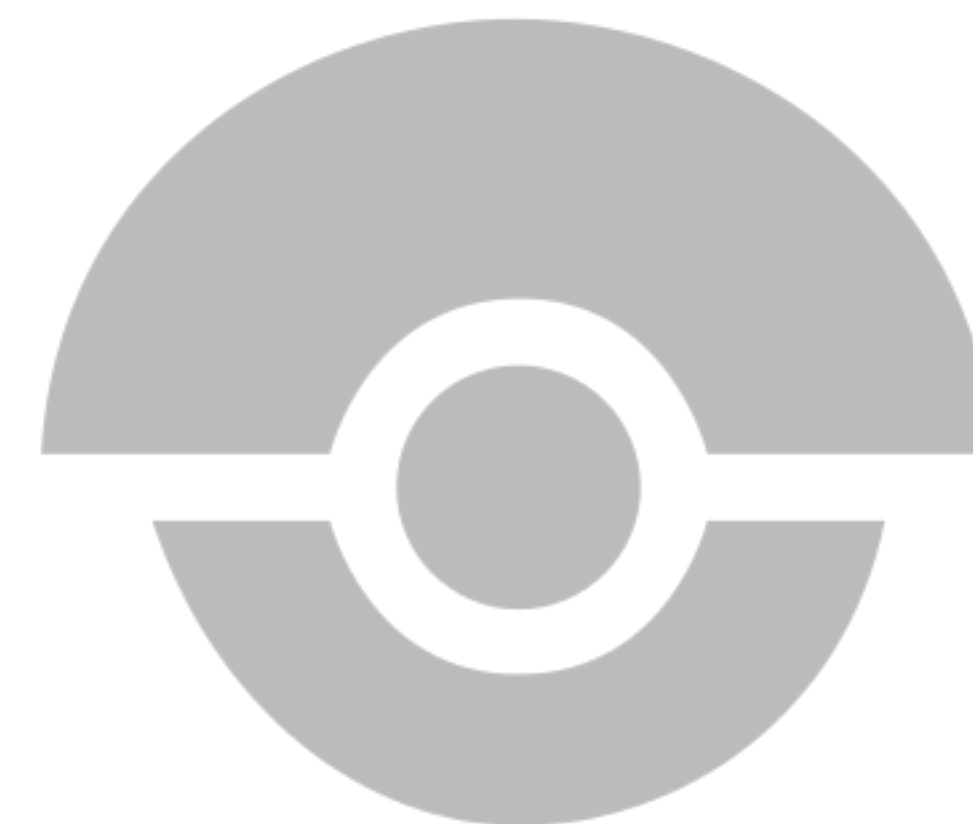
integrity



CODESHIP



Jenkins



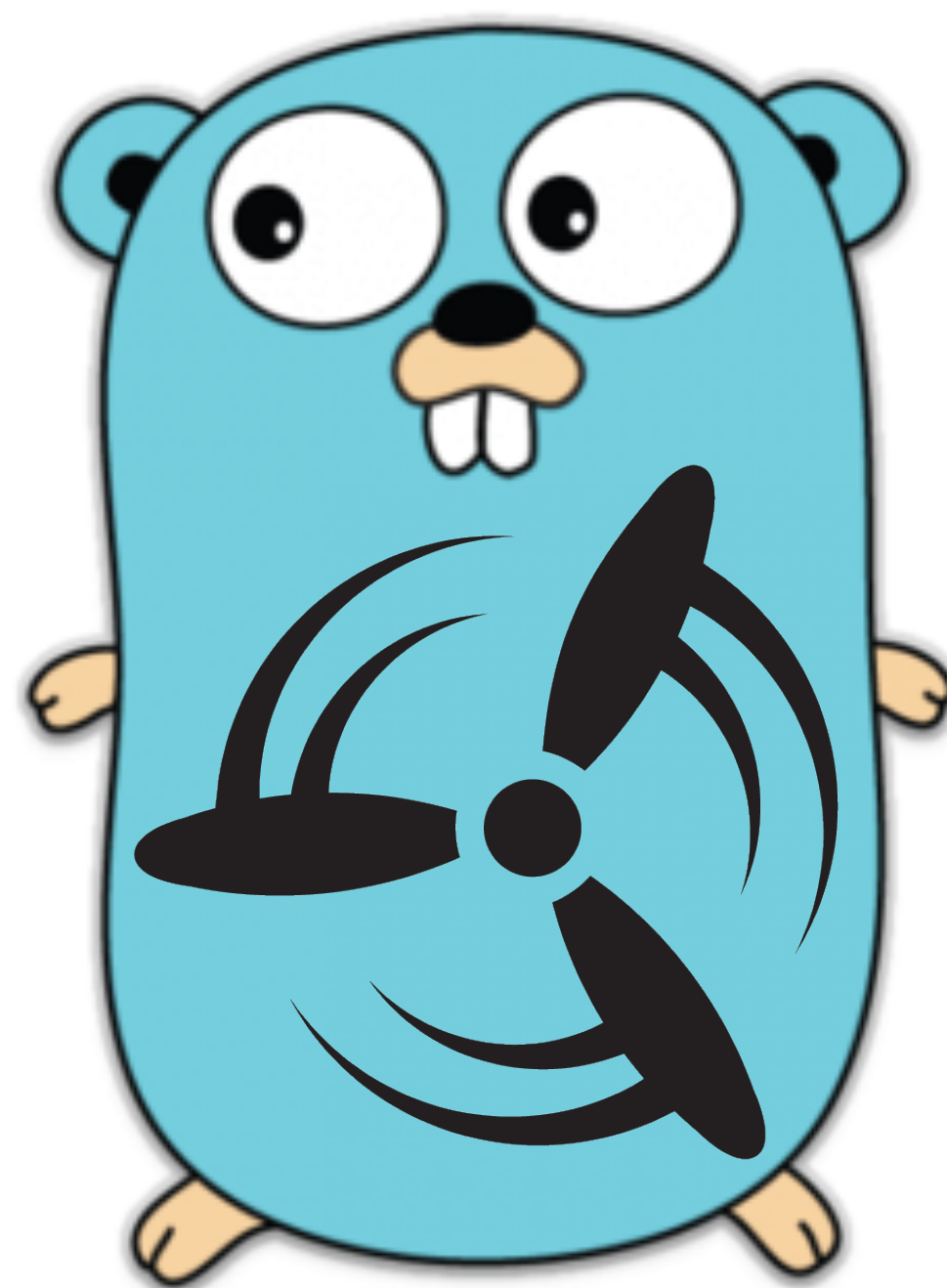
Travis CI



# О чем доклад?

- Как устроен Concourse CI
- Для чего CLI-утилита fly?
- Что делает Concourse CI в Devino Telecom?
- Что такое pipeline в Concourse?
- Для чего resource?
- Что происходит в job`е?





**Go**



**TSA**

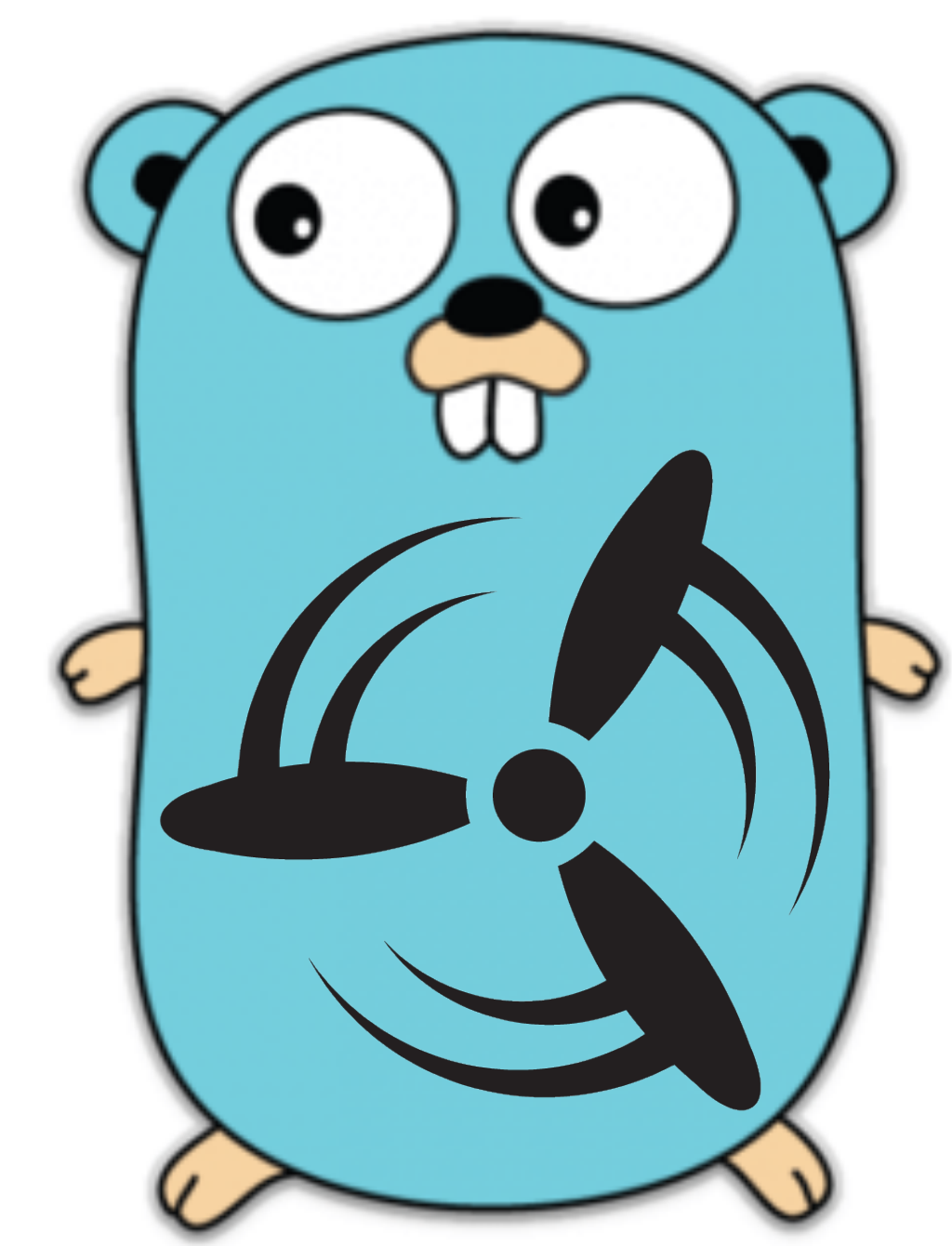
transportation security administration

Secure Worker Registration

**ATC**

air traffic control

Web UI, API, Pipeline Scheduler



**Go**



**TSA**

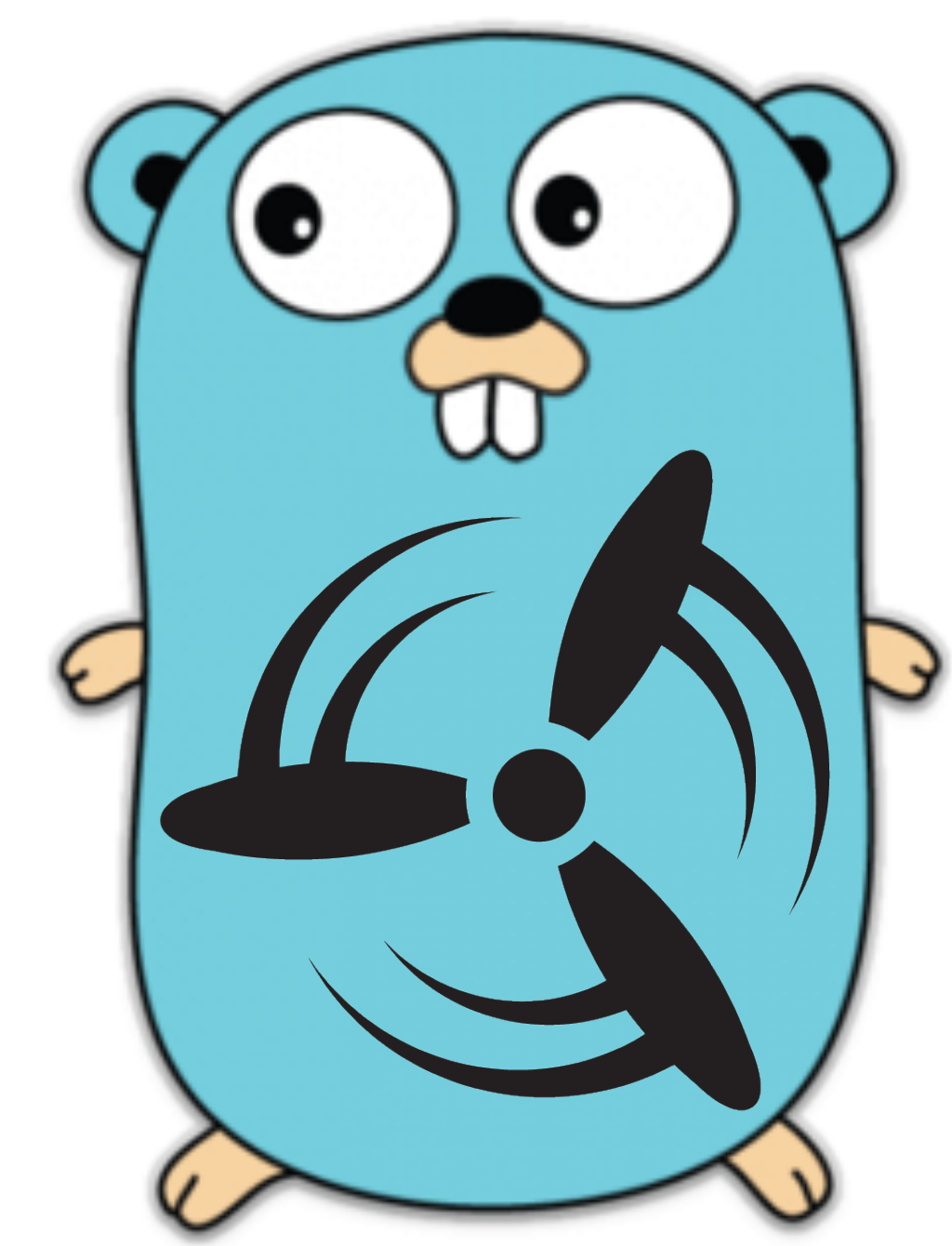
transportation security administration

Secure Worker Registration

**ATC**

air traffic control

Web UI, API, Pipeline Scheduler



**Go**



**TSA**

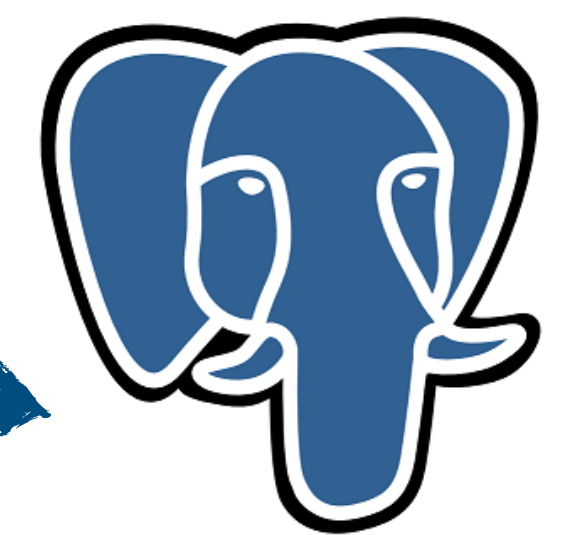
transportation security administration

Secure Worker Registration

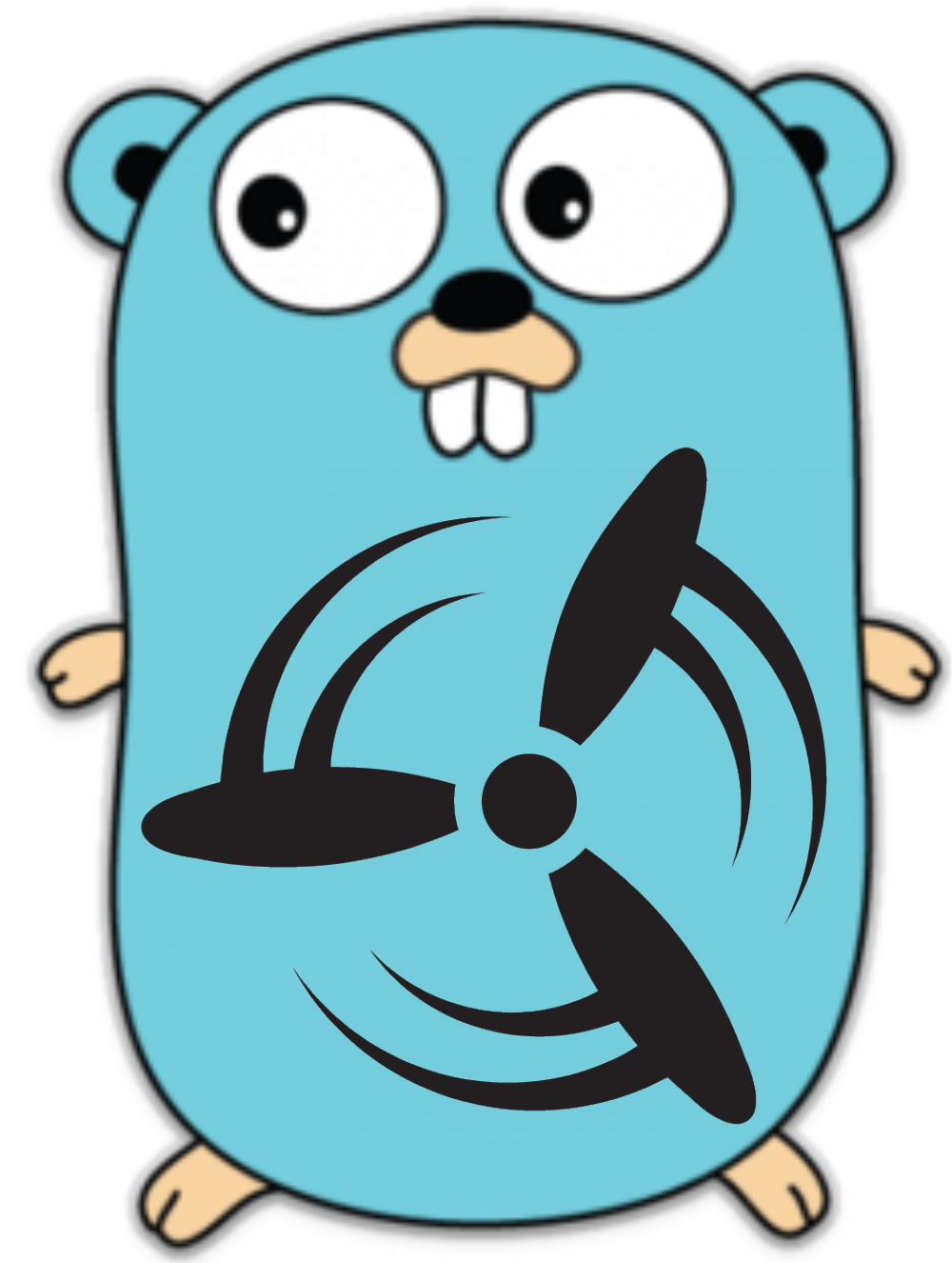
**ATC**

air traffic control

Web UI, API, Pipeline Scheduler



PostgreSQL



**Go**



**TSA**

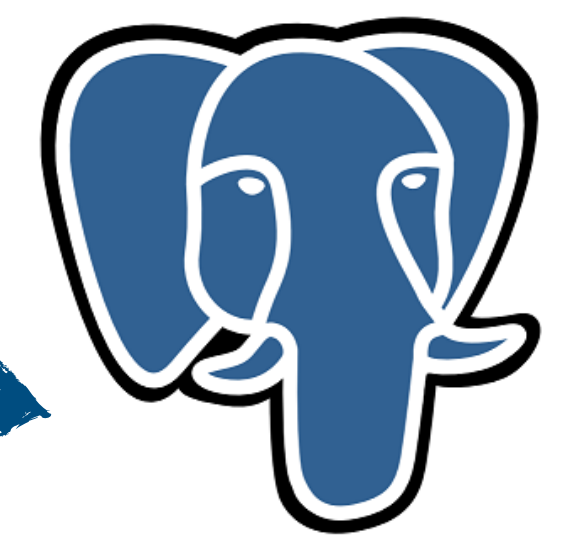
transportation security administration

Secure Worker Registration

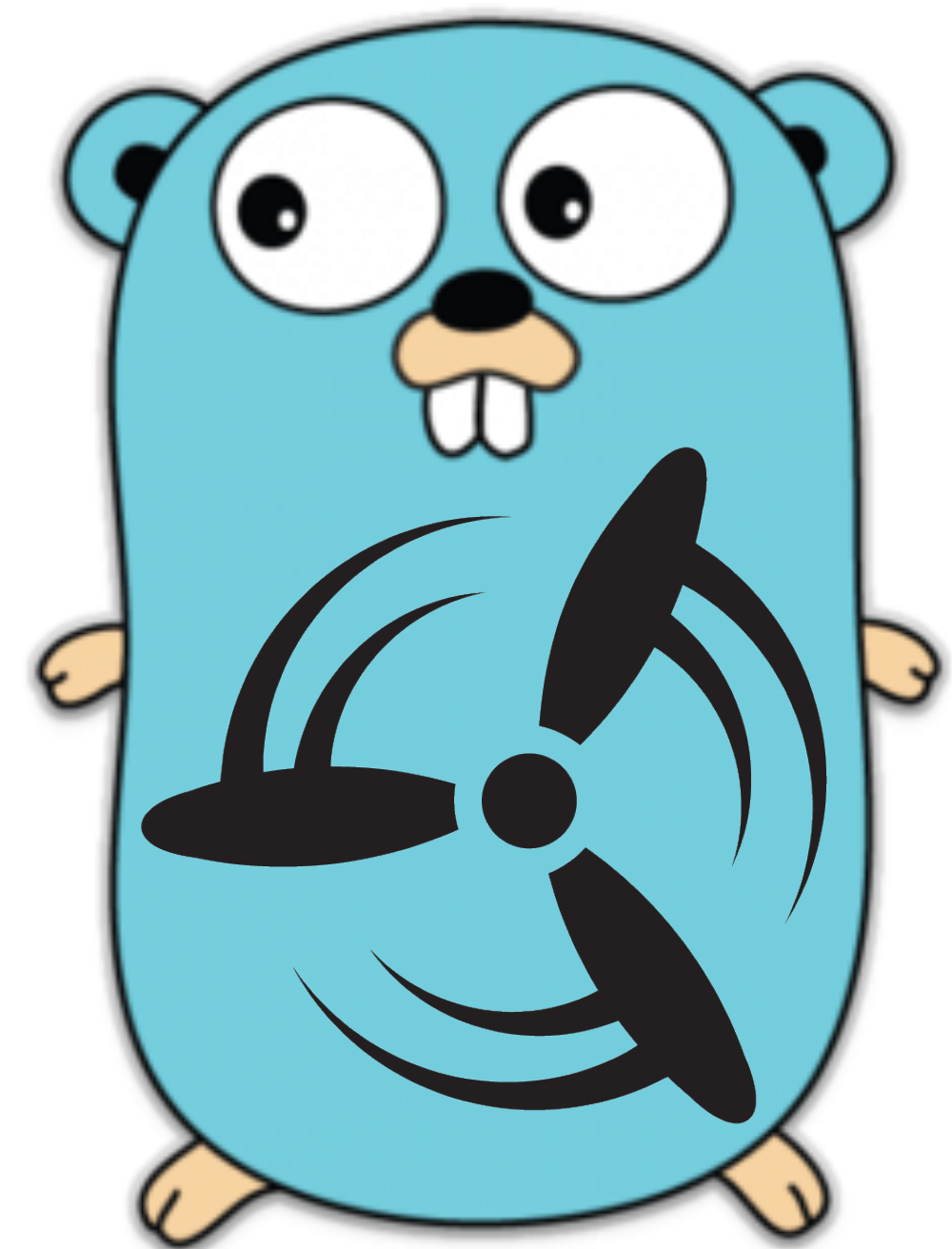
**ATC**

air traffic control

Web UI, API, Pipeline Scheduler



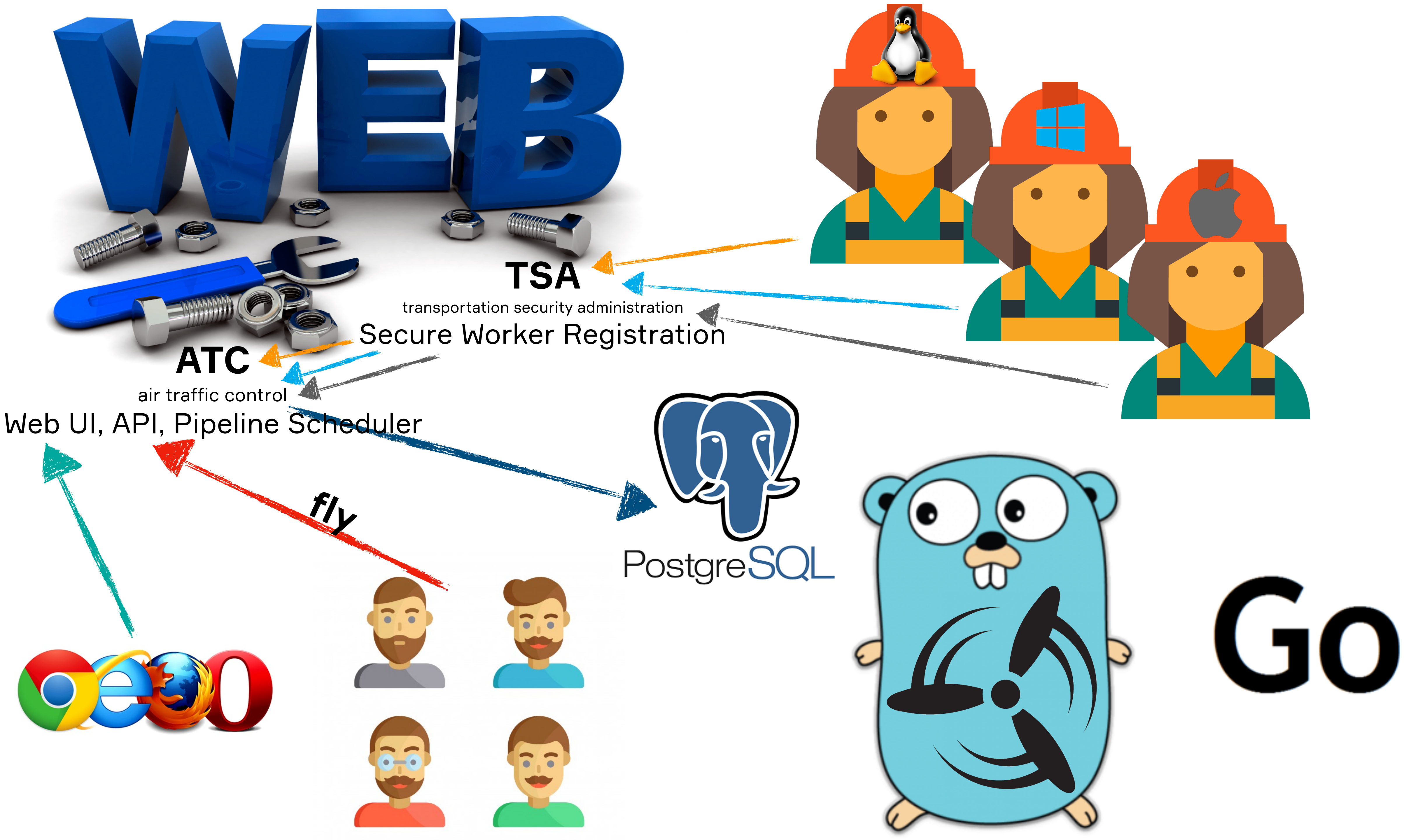
PostgreSQL



**Go**



*fly*



# Для чего CLI-утилита fly?

12

## ► Управление team`ами

```
$ fly -t main set-team --team-name=develop --ldap-group=...
```

## ► Управление pipeline`ами

```
$ fly -t develop --pipeline=HELLO-WORLD --config=pipelines/hello-world.yml
```

## ► Тестирование task`ов

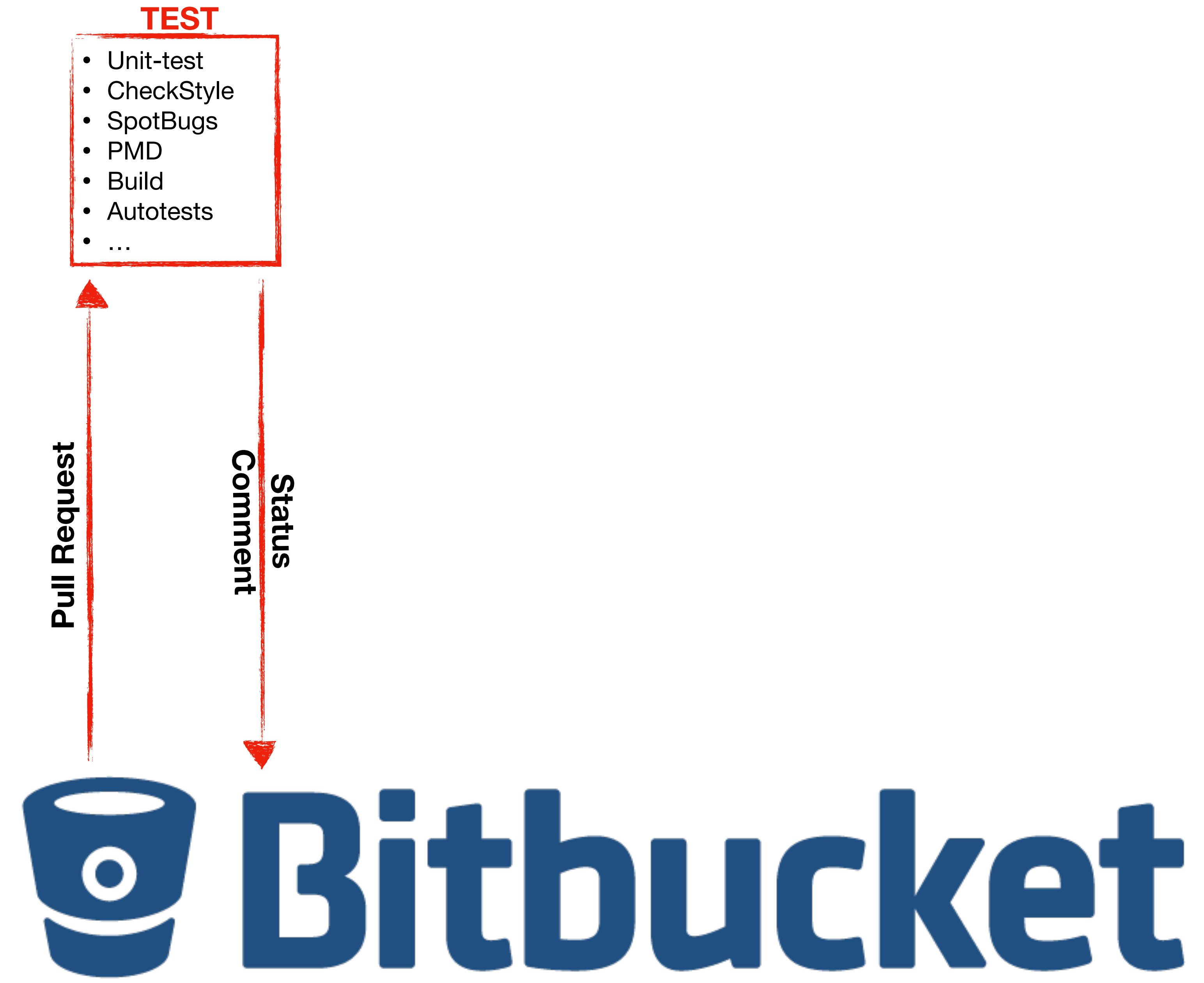
```
$ fly -t develop execute --config=ci/build.yml --input=source=.
```

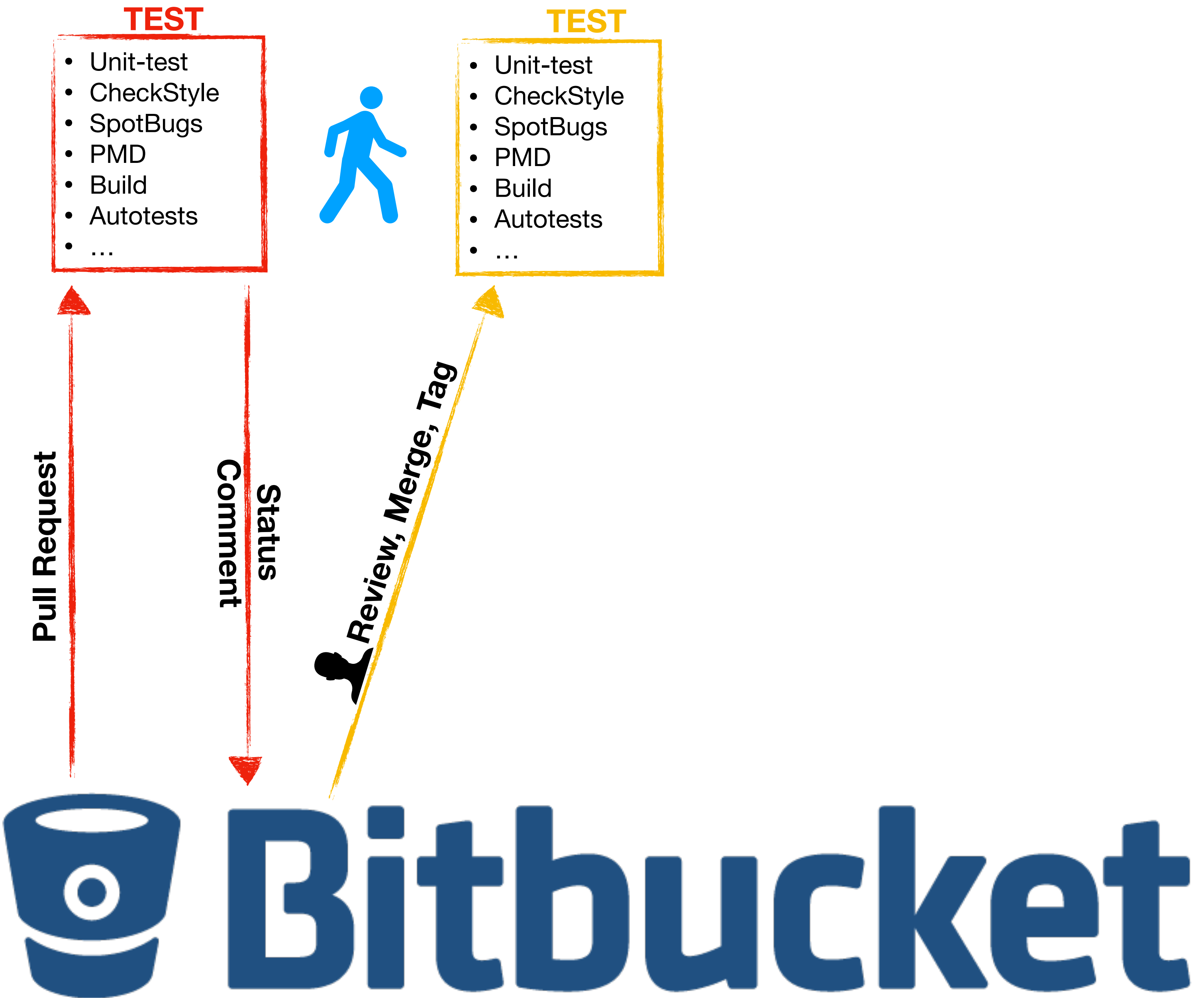
## ► Отладка работающего task`а

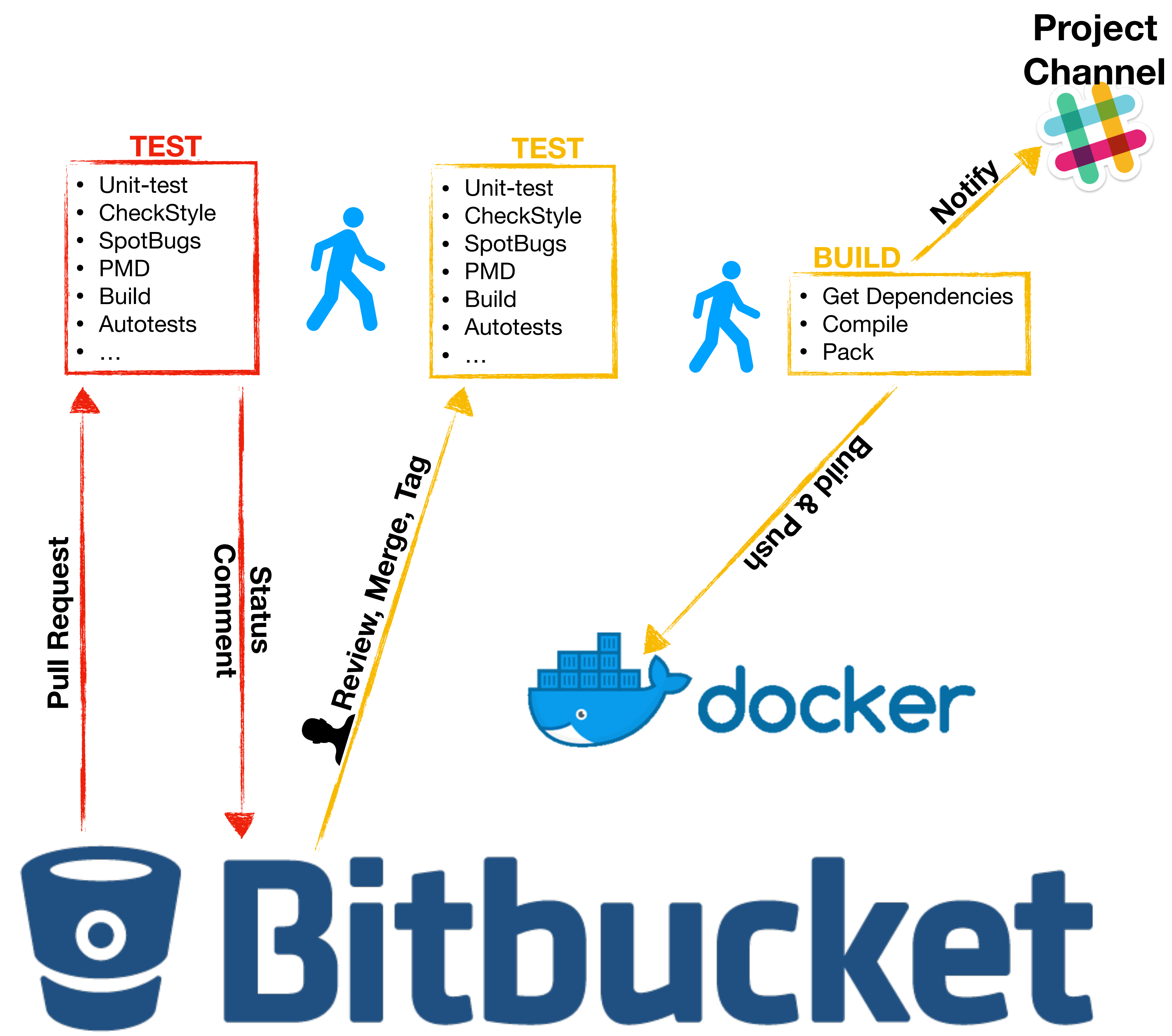
```
$ fly -t develop hijack --job=HELLO-WORLD/hello-world_build
```

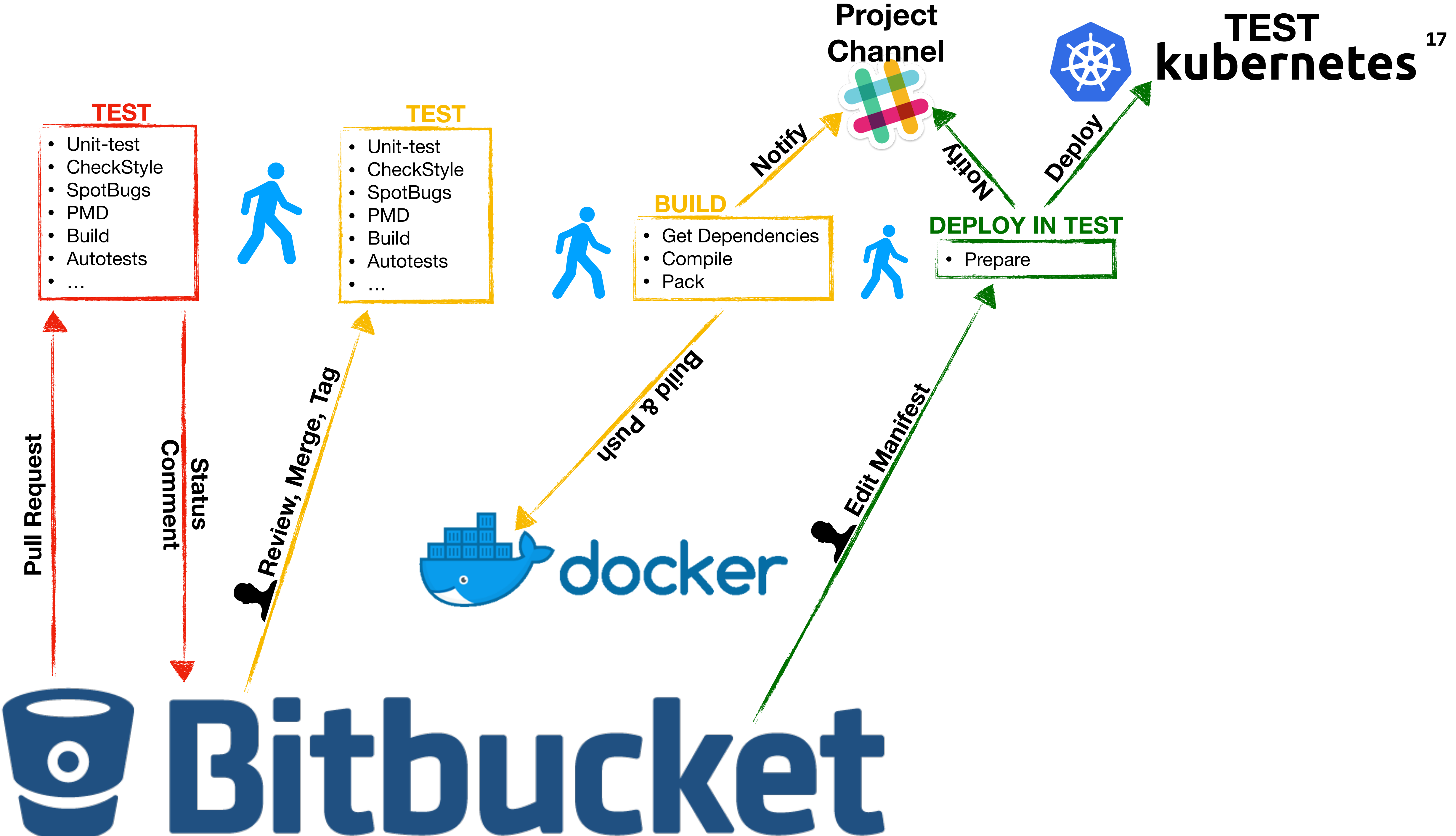
► ...

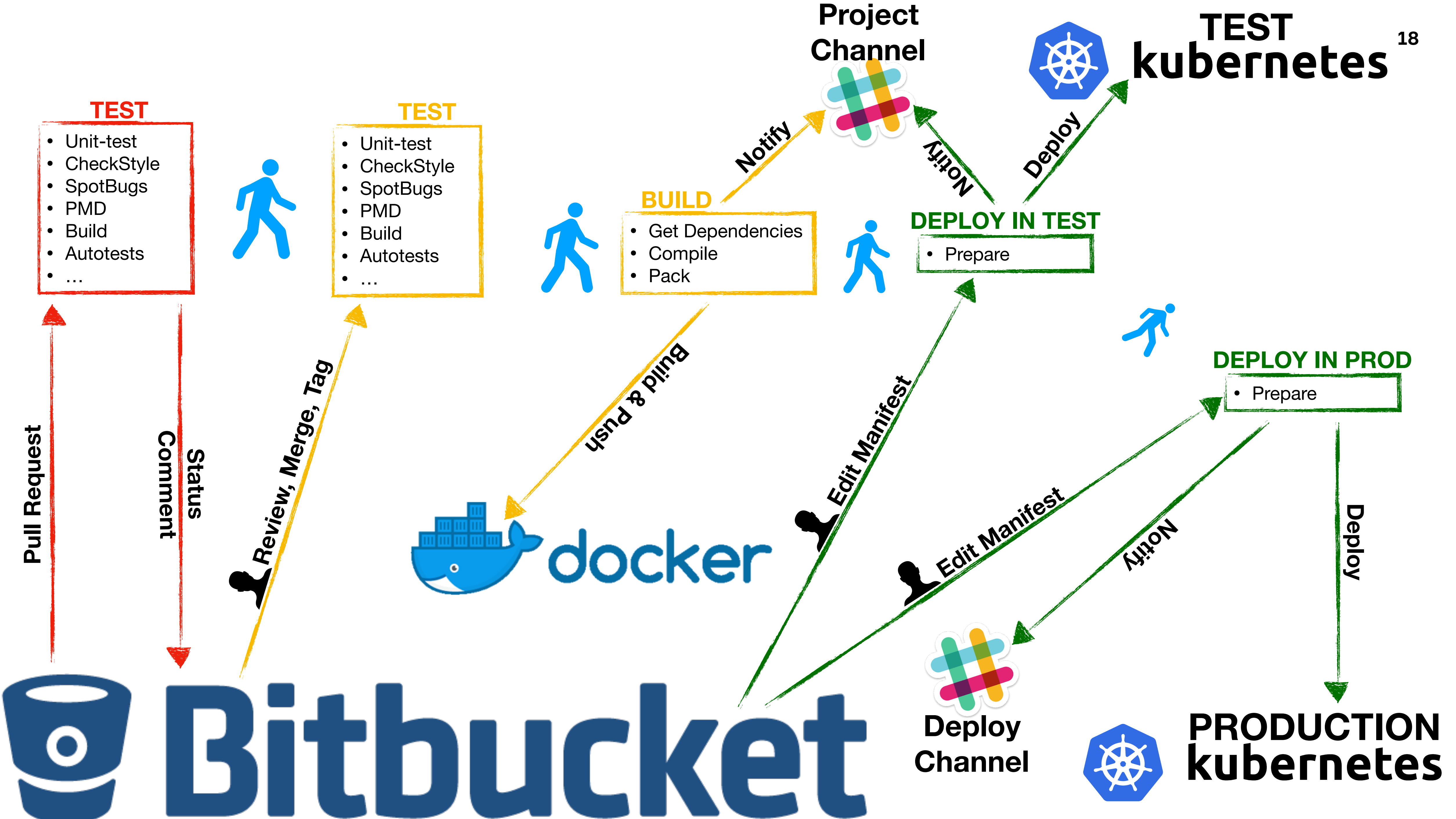
**Что делает Concourse CI в Devino Telecom?**

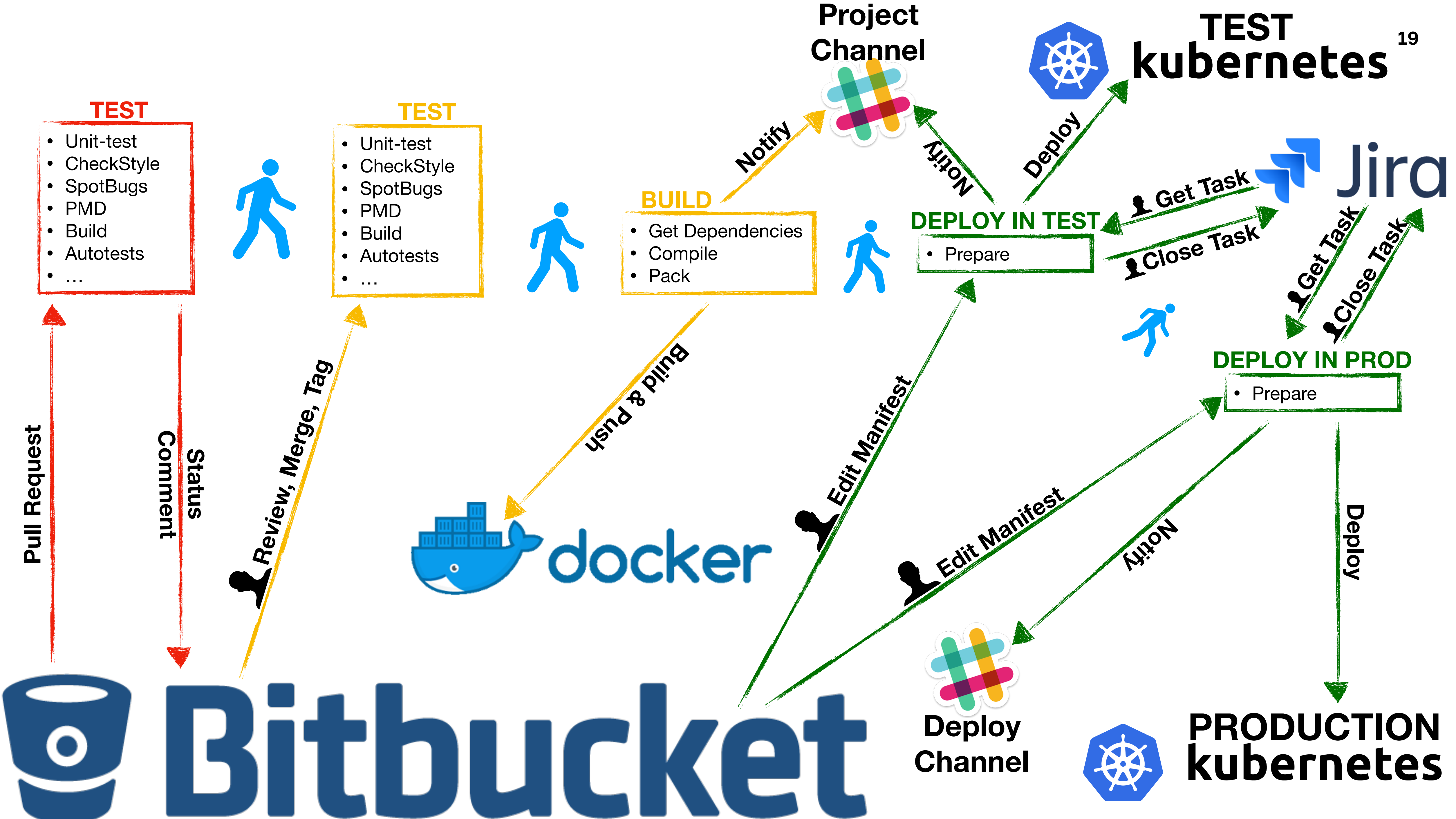






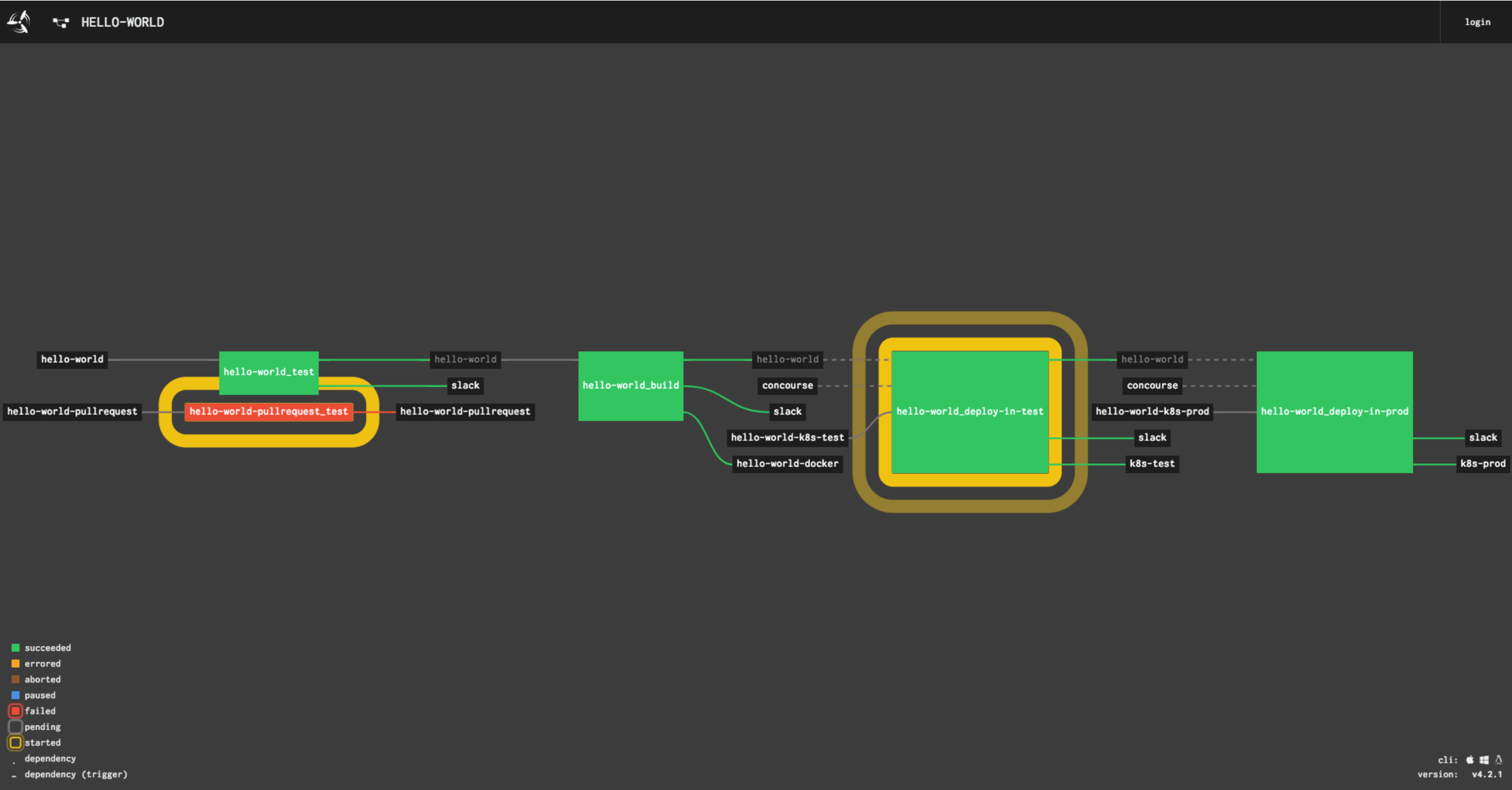




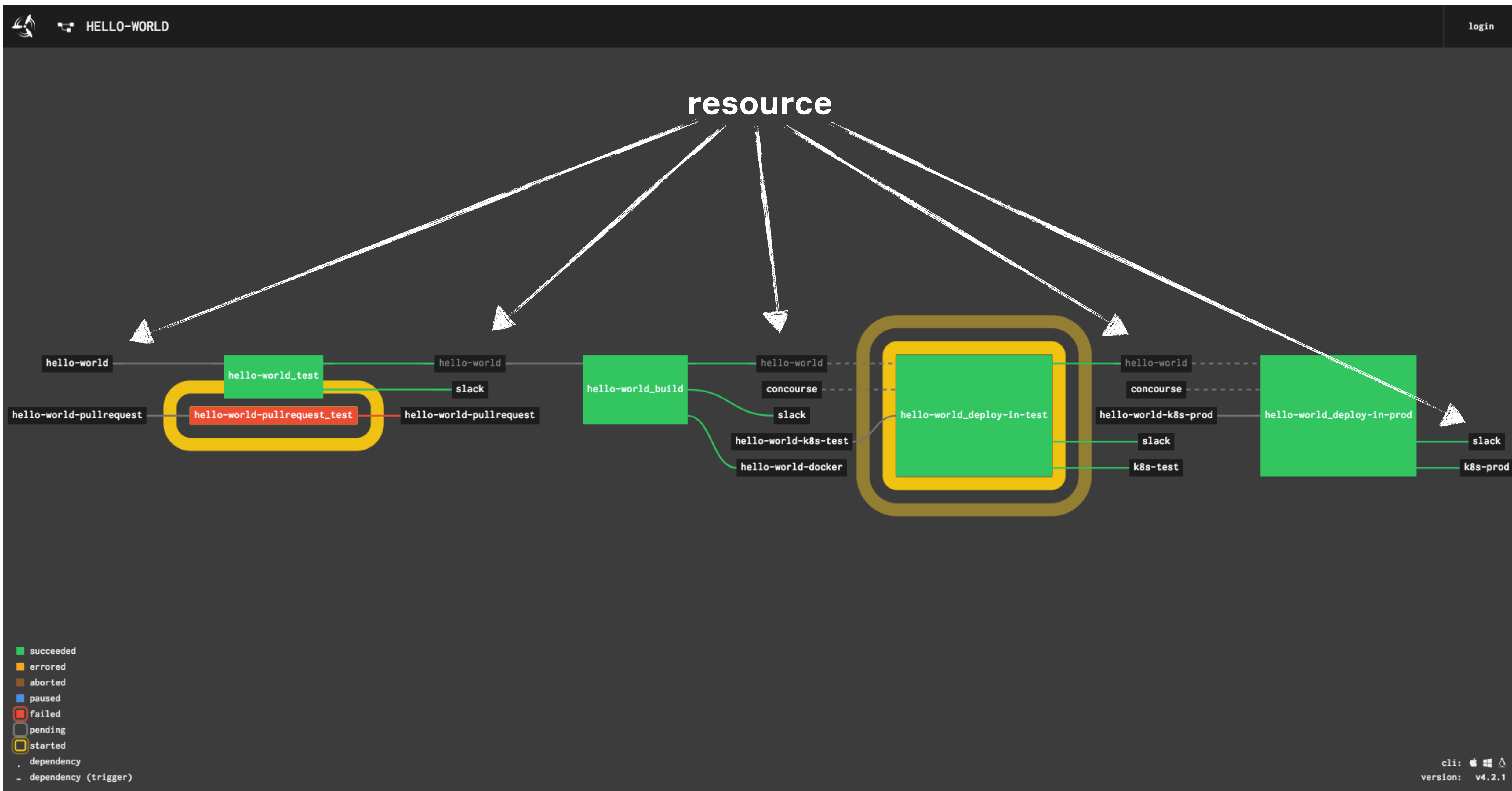


**Что такое pipeline в Concourse?**

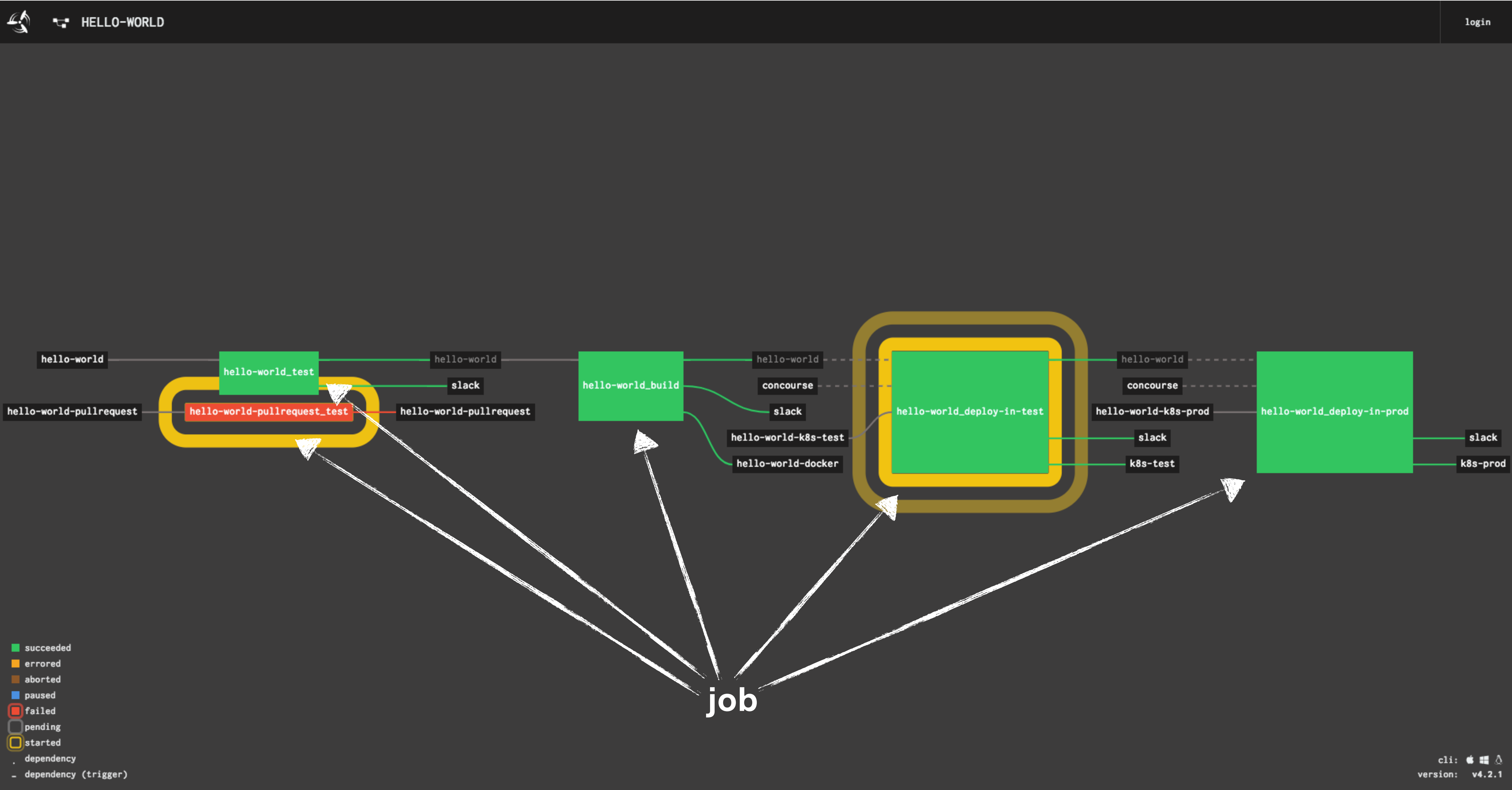
# Concourse CI v4.x.x - Pipeline



# Concourse CI v4.x.x - Pipeline



# Concourse CI v4.x.x - Pipeline



# Concourse CI v4.x.x - Main Dashboard



search

×

login

develop

HELLO-WORLD

3h 10m

FOO-BAR

8d 1h

HUNGRY-BERDEEN

paused

KIND-TURING

25d 2h

RELAXED-BOSE

21h 38m

ANGRY-JONES

1d 3h

HAPPY-MENDELEEV

11d 2h

EPIC-GOLDBERG

8d 3h

STOIC-ALLEN

10d 21h

PADANTIC-FERMI

41d 4h

CONFIDENT-BOHR

1d 4h

HOPEFUL-HOPPER

paused

PENDING

PAUSED

RUNNING

FAILING

ERRORED

ABORTED

SUCCEEDED

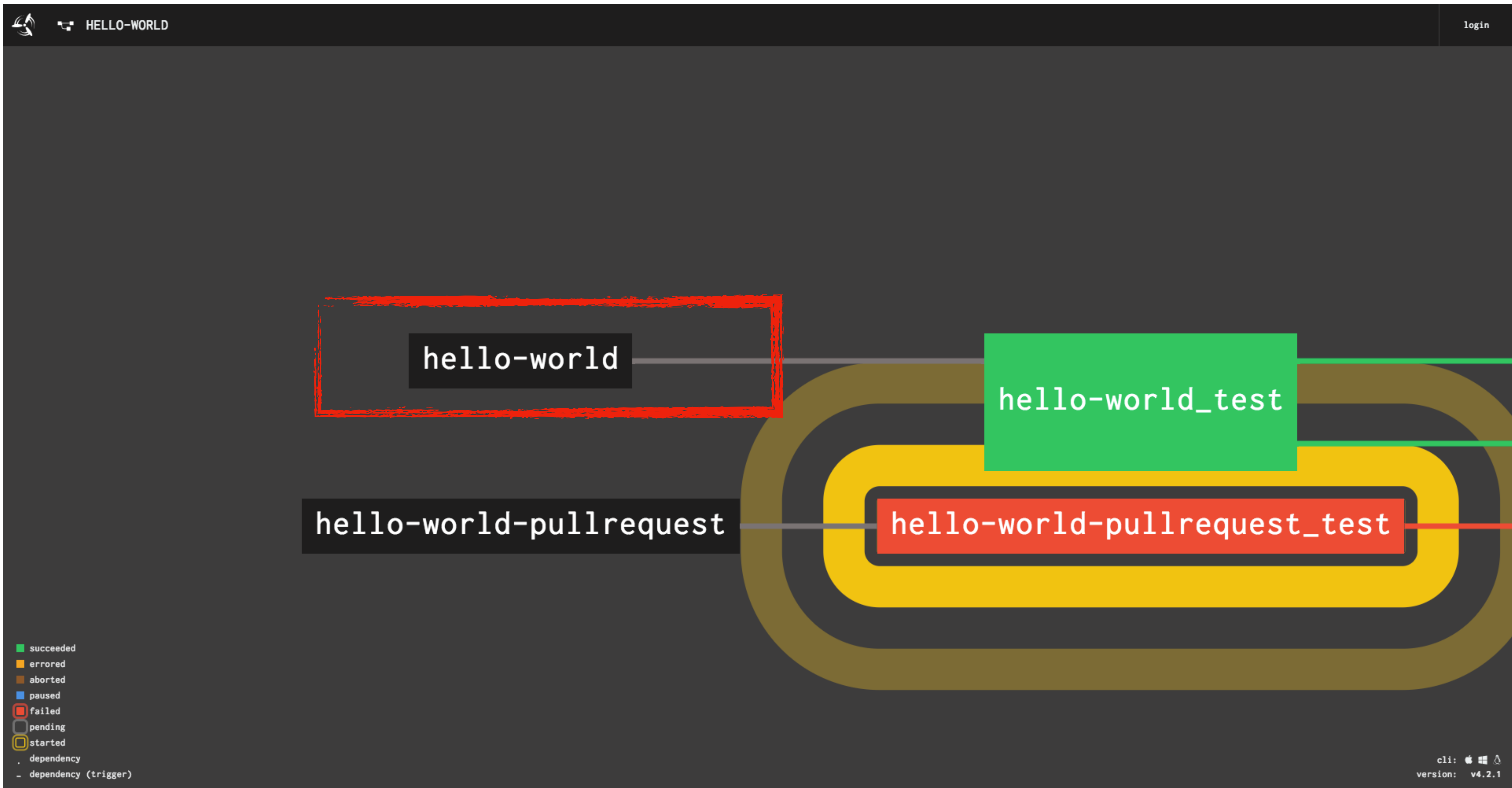
HIGH-DENSITY

version: v4.2.1

cli:   

**Для чего resource?**

# Concourse CI v4.x.x - Resource



# Concourse CI v4.x.x - Resource

27

A terminal window with a dark background. The text 'hello-world' is displayed in white. To the right of the text, there are three horizontal bars of increasing width, colored olive green, dark grey, and yellow from top to bottom. A thin vertical green bar is on the far right.

hello-world

**resources:**

- **name:** hello-world

**type:** git

**source:**

**uri:** ssh://git@git.example.com/ex/hello-world.git

**private\_key:** {{bitbucket-private-key}}

**branch:** master

**tag\_filter:** v\*

A terminal window with a dark background. The text 'hello-world' is displayed in white. To the right of the text is a vertical green bar. Below the text are two horizontal bars, one olive green and one yellow, both with rounded ends. A large, light gray 'YAML' watermark is visible in the background of the slide.

hello-world

**resources:**

- **name:** hello-world  
**type:** git

**source:**

**uri:** ssh://git@git.example.com/ex/hello-world.git

**private\_key:** {{bitbucket-private-key}}

**branch:** master

**tag\_filter:** v\*

# Concourse CI v4.x.x - Resource

29

A terminal window with a dark background. The text 'hello-world' is displayed in white. To the right of the text, there are three horizontal bars of increasing width, colored olive green, dark grey, and yellow from top to bottom. A thin vertical green bar is on the far right edge of the terminal window.

hello-world

**resources:**

- **name:** hello-world

**type:** git

**source:**

**uri:** ssh://git@git.example.com/ex/hello-world.git

**private\_key:** {{bitbucket-private-key}}

**branch:** master

**tag\_filter:** v\*

# Concourse CI v4.x.x - Resource

30

A terminal window with a dark background. The text 'hello-world' is displayed in white. To the right of the text, there are three horizontal bars of increasing width, colored olive green, dark grey, and yellow from top to bottom. A vertical green bar is on the far right edge of the terminal window.

hello-world

**resources:**

- **name:** hello-world

**type:** git

**source:**

**uri:** ssh://git@git.example.com/ex/hello-world.git

**private\_key:** |

-----BEGIN EC PRIVATE KEY-----

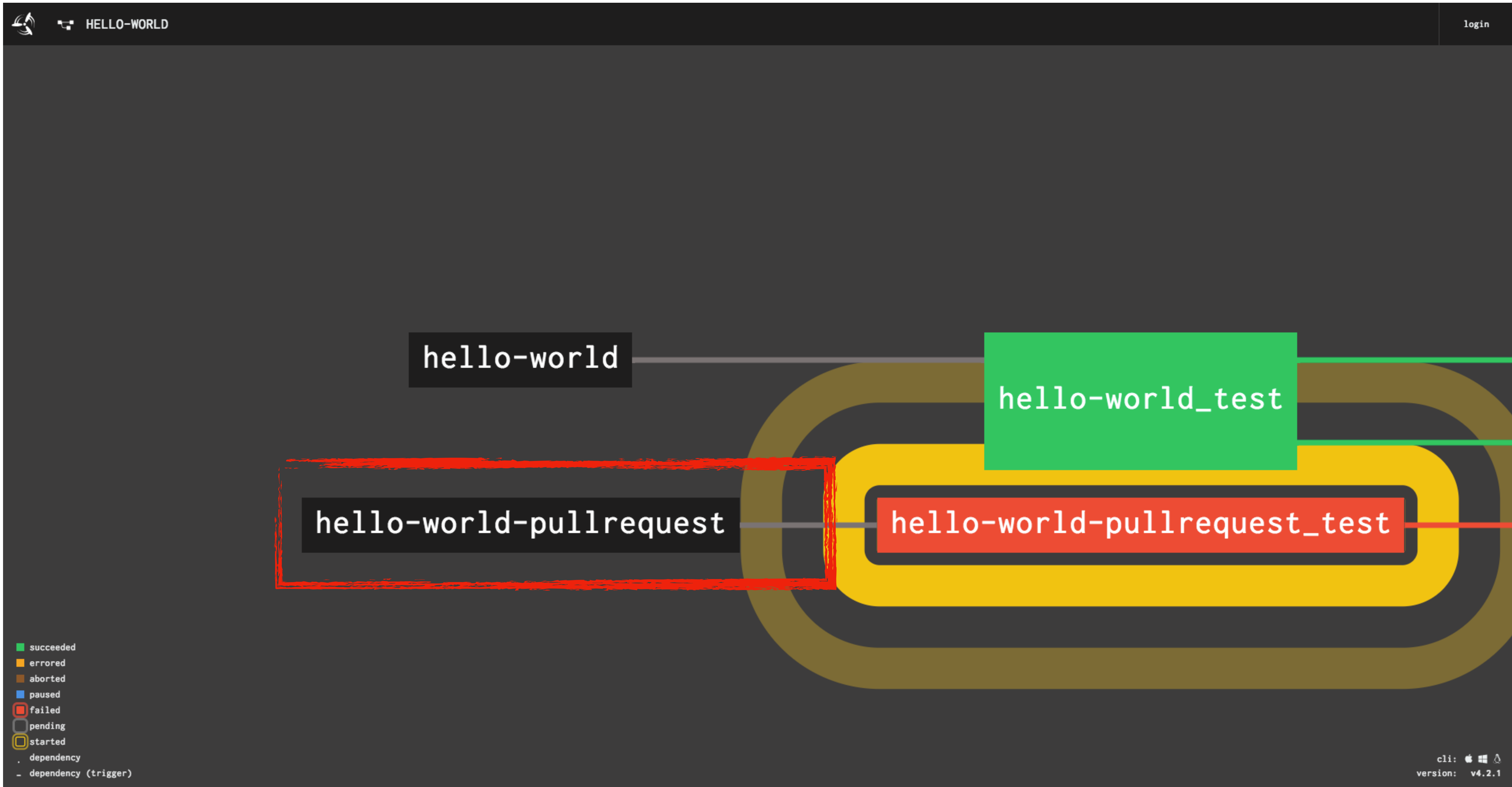
...

-----END EC PRIVATE KEY-----

**branch:** master

**tag\_filter:** v\*

# Concourse CI v4.x.x - Resource (community)





## resource\_types:

- **name:** bitbucket-pullrequest  
**type:** docker-image  
**source:** {repository: laurentverbruggen/concourse-bitbucket-pullrequest-resource}

## resources:

- **name:** hello-world-pullrequest  
**type:** bitbucket-pullrequest  
**source:**
  - uri:** https://git.example.com/scm/ex/hello-world.git
  - username:** {{bitbucket-username}}
  - password:** {{bitbucket-password}}
  - only\_for\_branch:** master
  - only\_without\_conflicts:** true
  - rebuild\_when\_target\_changed:** true



## resource\_types:

- **name:** bitbucket-pullrequest  
**type:** docker-image  
**source:** {repository: laurentverbruggen/concourse-bitbucket-pullrequest-resource}

## resources:

- **name:** hello-world-pullrequest  
**type:** bitbucket-pullrequest  
**source:**
  - uri:** https://git.example.com/scm/ex/hello-world.git
  - username:** {{bitbucket-username}}
  - password:** {{bitbucket-password}}
  - only\_for\_branch:** master
  - only\_without\_conflicts:** true
  - rebuild\_when\_target\_changed:** true



## resource\_types:

- **name:** bitbucket-pullrequest

**type:** docker-image

**source:** {repository: laurentverbruggen/concourse-bitbucket-pullrequest-resource}

## resources:

- **name:** hello-world-pullrequest

**type:** bitbucket-pullrequest

### source:

**uri:** https://git.example.com/scm/ex/hello-world.git

**username:** {{bitbucket-username}}

**password:** {{bitbucket-password}}

**only\_for\_branch:** master

**only\_without\_conflicts:** true

**rebuild\_when\_target\_changed:** true



## resource\_types:

- **name:** bitbucket-pullrequest  
**type:** docker-image  
**source:** {repository: laurentverbruggen/concourse-bitbucket-pullrequest-resource}

## resources:

- **name:** hello-world-pullrequest  
**type:** bitbucket-pullrequest  
**source:**
  - uri:** https://git.example.com/scm/ex/hello-world.git
  - username:** {{bitbucket-username}}
  - password:** {{bitbucket-password}}
  - only\_for\_branch:** master
  - only\_without\_conflicts:** true
  - rebuild\_when\_target\_changed:** true



## resource\_types:

- **name:** bitbucket-pullrequest  
**type:** docker-image  
**source:** {repository: laurentverbruggen/concourse-bitbucket-pullrequest-resource}

## resources:

- **name:** hello-world-pullrequest  
**type:** bitbucket-pullrequest  
**source:**
  - uri:** https://git.example.com/scm/ex/hello-world.git
  - username:** {{bitbucket-username}}
  - password:** {{bitbucket-password}}
  - only\_for\_branch:** master
  - only\_without\_conflicts:** true
  - rebuild\_when\_target\_changed:** true

# Concourse CI v4.x.x - Resource (community)

37



## resource\_types:

- **name:** bitbucket-pullrequest  
**type:** docker-image  
**source:** {repository: laurentverbruggen/concourse-bitbucket-pullrequest-resource}

## resources:

- **name:** hello-world-pullrequest  
**type:** bitbucket-pullrequest  
**source:**
  - uri:** https://git.example.com/scm/ex/hello-world.git
  - username:** concourse
  - password:** #sEcRt\_pAss-&1
  - only\_for\_branch:** master
  - only\_without\_conflicts:** true
  - rebuild\_when\_target\_changed:** true

# Concourse CI v4.x.x - Resource Types

38

## Provided With Concourse

- The `git` resource can pull and push to git repositories.
- The `hg` resource can pull and push to Mercurial repositories.
- The `time` resource can start jobs on a schedule or timestamp outputs.
- The `s3` resource can fetch from and upload to S3 buckets.
- The `archive` resource can fetch and extract .tar.gz archives.
- The `semver` resource can set or bump version numbers.
- The `github-release` resource can fetch and publish versioned GitHub resources.
- The `docker-image` resource can fetch, build, and push Docker images
- The `tracker` resource can deliver stories and bugs on Pivotal Tracker
- The `pool` resource allows you to configure how to serialize use of an external system. This lets you prevent test interference or overwork on shared systems.
- The `cf` resource can deploy an application to Cloud Foundry.
- The `bosh-io-release` resource can track and fetch new BOSH releases from bosh.io.
- The `bosh-io-stemcell` resource can track and fetch new BOSH stemcells from bosh.io.

# Concourse CI v4.x.x - Resource Types

39

## Provided With Concourse

- The `git` resource can pull and push to git repositories.
- The `hg` resource can pull and push to Mercurial repositories.
- The `time` resource can start jobs on a schedule or timestamp outputs.
- The `s3` resource can fetch from and upload to S3 buckets.
- The `archive` resource can fetch and extract .tar.gz archives.
- The `semver` resource can set or bump version numbers.
- The `github-release` resource can fetch and publish versioned GitHub resources.
- The `docker-image` resource can fetch, build, and push Docker images
- The `tracker` resource can deliver stories and bugs on Pivotal Tracker
- The `pool` resource allows you to configure how to serialize use of an external system. This lets you prevent test interference or overwork on shared systems.
- The `cf` resource can deploy an application to Cloud Foundry.
- The `bosh-io-release` resource can track and fetch new BOSH releases from bosh.io.
- The `bosh-io-stemcell` resource can track and fetch new BOSH stemcells from bosh.io.

## Community

### Resource Name

[Slack Reading and Posting](#)

[Slack notifications](#)

[Github Pull Requests](#)

[GitLab Merge Requests](#)

[OpenStack Swift](#)

[Key Value resource](#)

[Flowdock notifications](#)

[Email](#)

[Formatted Email](#)

[Email with integrated MTA](#)

[Bintray](#)

[Perforce](#)

[BOSH Errands](#)

[BOSH Config: cloud-config and runtime-config](#)

[BOSH Release](#)

[Pool Trigger](#)

[Pivotal Network](#)

[FTP](#)

[lftp, access resources via ftp, http, sftp and fish](#)

[Cloudformation](#)

[Generic HTTP API](#)

[Hockey App](#)

[Concourse Pipelines](#)

### Maintained By

by [@jleben](#)

by [@cloudfoundry-community](#)

by [@telia-oss](#)

by [@swisscom](#)

by [@sapcc](#)

by [@swce](#)

by [@starkandwayne](#)

by [@pivotal-cf](#)

by [@santoshjpawar](#)

by [@mdomke](#)

by [@jamiemonserate](#)

by [@olhtbr](#)

by [@starkandwayne](#)

by [@EMC-Dojo](#)

by [@dpb587](#)

by [@sfmobile](#)

by [@pivotal-cf-experimental](#)

by [@aequitas](#)

by [@openSUSE](#)

by [@pivotal-cf-experimental](#)

by [@aequitas](#)

by [@seadowg](#)

by [@robdimsdale](#)

# Concourse CI v4.x.x - Resource Types

## Provided With Concourse

- The `git` resource can pull and push to git repositories.
- The `hg` resource can pull and push to Mercurial repositories.
- The `time` resource can start jobs on a schedule or timestamp outputs.
- The `s3` resource can fetch from and upload to S3 buckets.
- The `archive` resource can fetch and extract .tar.gz archives.
- The `semver` resource can set or bump version numbers.
- The `github-release` resource can fetch and publish versioned GitHub resources.
- The `docker-image` resource can fetch, build, and push Docker images
- The `tracker` resource can deliver stories and bugs on Pivotal Tracker
- The `pool` resource allows you to configure how to serialize use of an external system. This lets you prevent test interference or overwork on shared systems.
- The `cf` resource can deploy an application to Cloud Foundry.
- The `bosh-io-release` resource can track and fetch new BOSH releases from bosh.io.
- The `bosh-io-stemcell` resource can track and fetch new BOSH stemcells from bosh.io.

## Community

| Resource Name                             | Maintained By                              |
|-------------------------------------------|--------------------------------------------|
| <a href="#">Slack Reading and Posting</a> | by <a href="#">@jleben</a>                 |
| <a href="#">Slack notifications</a>       | by <a href="#">@cloudfoundry-community</a> |
| <a href="#">Github Pull Requests</a>      | by <a href="#">@telia-oss</a>              |
| <a href="#">GitLab Merge Requests</a>     | by <a href="#">@swisscom</a>               |
| <a href="#">OpenStack Swift</a>           | by <a href="#">@sapcc</a>                  |
| <a href="#">Key Value resource</a>        | by <a href="#">@swce</a>                   |
| <a href="#">Flowdock notifications</a>    | by <a href="#">@starkandwayne</a>          |
| <a href="#">Email</a>                     | by <a href="#">@pivotal-cf</a>             |
| <a href="#">Formatted Email</a>           | by <a href="#">@santoshjpawar</a>          |

<https://concourse-ci.org/community-resources.html>

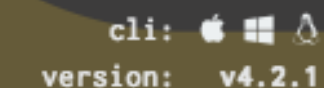
|                                                                     |                                             |
|---------------------------------------------------------------------|---------------------------------------------|
| <a href="#">Perforce</a>                                            | by <a href="#">@olhtbr</a>                  |
| <a href="#">BOSH Errands</a>                                        | by <a href="#">@starkandwayne</a>           |
| <a href="#">BOSH Config: cloud-config and runtime-config</a>        | by <a href="#">@EMC-Dojo</a>                |
| <a href="#">BOSH Release</a>                                        | by <a href="#">@dpb587</a>                  |
| <a href="#">Pool Trigger</a>                                        | by <a href="#">@sfmobile</a>                |
| <a href="#">Pivotal Network</a>                                     | by <a href="#">@pivotal-cf-experimental</a> |
| <a href="#">FTP</a>                                                 | by <a href="#">@aequitas</a>                |
| <a href="#">lftp, access resources via ftp, http, sftp and fish</a> | by <a href="#">@openSUSE</a>                |
| <a href="#">Cloudformation</a>                                      | by <a href="#">@pivotal-cf-experimental</a> |
| <a href="#">Generic HTTP API</a>                                    | by <a href="#">@aequitas</a>                |
| <a href="#">Hockey App</a>                                          | by <a href="#">@seadowg</a>                 |
| <a href="#">Concourse Pipelines</a>                                 | by <a href="#">@robdimsdale</a>             |

```
$ docker run --rm --entrypoint ls devinotelecom/concourse-slack-alert-resource -l /opt/resource
total 19340
-rwxr-xr-x 1 root root 6451874 Nov 1 09:57 check
-rwxr-xr-x 1 root root 6452080 Nov 1 09:57 in
-rwxr-xr-x 1 root root 6879238 Nov 1 09:57 out
```

<https://concourse-ci.org/implementing-resources.html>

**Что происходит в job`e?**

## 43



# Concourse CI v4.x.x - Job "\*\_build"

44

The screenshot displays the Concourse CI web interface for a job named 'hello-world\_build'. The top navigation bar shows 'HELLO-WORLD / <> hello-world\_build' and a 'login' link. Below this, a green header bar indicates the job is 'hello-world\_build #8', with details: 'started 3m 49s ago', 'finished 2m 56s ago', and 'duration 53s'. A progress bar below the header shows 8 steps, with steps 1 through 5 highlighted in orange and steps 6 through 8 in green. The main content area shows a tree view of the job steps: 'hello-world' (with a 'ref' and a green checkmark), '>\_ build' (with a green checkmark), and 'hello-world-docker' (with a 'digest' and a green checkmark). The 'hello-world-docker' step is expanded, showing sub-steps: 'hello-world-docker' (with a 'digest' and a green checkmark), 'slack' (with a green dot), 'slack' (with a green dot), 'slack' (with a green checkmark and 'ver static'), and 'slack' (with a green checkmark and 'ver static').

```
jobs:
- name: hello-world_build
 plan:
 - get: hello-world
 passed: ["hello-world_test"]
 trigger: true
 - task: build
 file: hello-world/ci/build.yml
 input_mapping: {source: hello-world}
 - put: hello-world-docker
 params: {build: docker, tag: docker/version, tag_as_latest: true}
 on_failure:
 put: slack
 params: {alert_type: failed, message: "Build FAILED"}
 on_success:
 put: slack
 params: {alert_type: success, message: "Build SUCCESS"}
```

# Concourse CI v4.x.x - Job "\*\_build"

45

The screenshot shows the Concourse CI web interface. At the top, the breadcrumb navigation is 'HELLO-WORLD / <> hello-world\_build'. A 'login' link is in the top right. Below the breadcrumb, a green bar displays 'hello-world\_build #8' with a '+' icon on the right. To the right of the job name, it shows 'started 3m 49s ago', 'finished 2m 56s ago', and 'duration 53s'. Below this is a row of colored squares representing job history: 8 (green), 7 (green), 6 (green), 5 (green), 4 (green), 3 (orange), 2 (red), and 1 (orange). The main content area shows a tree view of the job steps: 'hello-world' (expanded), '>\_ build', 'hello-world-docker' (expanded), 'hello-world-docker', 'slack', 'slack', 'slack', and 'slack'. Each step has a status icon (down arrow for get/put, up arrow for task) and a green checkmark. The 'hello-world-docker' step shows a 'digest' and 'sha256' hash. The 'slack' steps show 'ver static' and a green checkmark. A large 'YAML' watermark is visible in the background.

```
jobs:
- name: hello-world_build
 plan:
 - get: hello-world
 passed: ["hello-world_test"]
 trigger: true
 - task: build
 file: hello-world/ci/build.yml
 input_mapping: {source: hello-world}
 - put: hello-world-docker
 params: {build: docker, tag: docker/version, tag_as_latest: true}
 on_failure:
 put: slack
 params: {alert_type: failed, message: "Build FAILED"}
 on_success:
 put: slack
 params: {alert_type: success, message: "Build SUCCESS"}
```

# Concourse CI v4.x.x - Job "\*\_build"

46

The screenshot shows the Concourse CI web interface for a job named 'hello-world\_build'. The top navigation bar includes the Concourse logo, the path 'HELLO-WORLD / <> hello-world\_build', and a 'login' link. Below the navigation bar, a green header bar displays 'hello-world\_build #8' along with its status: 'started 3m 49s ago', 'finished 2m 56s ago', and 'duration 53s'. A progress bar below the header shows 8 steps, with steps 1 through 5 highlighted in orange and steps 6 through 8 in green. The main content area lists the job steps: 'hello-world' (with a 'ref' and a green checkmark), '>\_ build' (with a green checkmark), 'hello-world-docker' (with a 'digest' and a green checkmark), and a nested list of 'hello-world-docker', 'slack', and 'slack' steps, each with a 'digest' and a green checkmark. The 'slack' steps are also marked with 'ver static' and a green checkmark.

```
jobs:
- name: hello-world_build
 plan:
 - get: hello-world
 passed: ["hello-world_test"]
 trigger: true
 - task: build
 file: hello-world/ci/build.yml
 input_mapping: {source: hello-world}
 - put: hello-world-docker
 params: {build: docker, tag: docker/version, tag_as_latest: true}
 on_failure:
 put: slack
 params: {alert_type: failed, message: "Build FAILED"}
 on_success:
 put: slack
 params: {alert_type: success, message: "Build SUCCESS"}
```

# Concourse CI v4.x.x - Job "\*\_build"

47

The screenshot shows the Concourse CI web interface. At the top, the breadcrumb navigation is 'HELLO-WORLD / <> hello-world\_build'. A 'login' link is in the top right. Below the breadcrumb, a green bar displays 'hello-world\_build #8' with a '+' icon on the right. To the right of this bar, the job status is shown: 'started 3m 49s ago', 'finished 2m 56s ago', and 'duration 53s'. Below the green bar is a row of colored squares representing the job's history, with '8' highlighted in green. The main content area shows a tree view of the job's steps: 'hello-world' (with a 'ref' and a green checkmark), '>\_ build' (with a green checkmark), and 'hello-world-docker' (with a green checkmark). The 'hello-world-docker' step is expanded, showing a sub-step 'hello-world-docker' (with a green checkmark) and two 'slack' steps (each with a green checkmark). The 'hello-world-docker' step also shows a 'digest' and a 'sha256' hash. The 'slack' steps show 'ver static' and a green checkmark.

```
jobs:
- name: hello-world_build
 plan:
 - get: hello-world
 passed: ["hello-world_test"]
 trigger: true
 - task: build
 file: hello-world/ci/build.yml
 input_mapping: {source: hello-world}
 - put: hello-world-docker
 params: {build: docker, tag: docker/version, tag_as_latest: true}
 on_failure:
 put: slack
 params: {alert_type: failed, message: "Build FAILED"}
 on_success:
 put: slack
 params: {alert_type: success, message: "Build SUCCESS"}
```

# Concourse CI v4.x.x - Task ci/build.yml

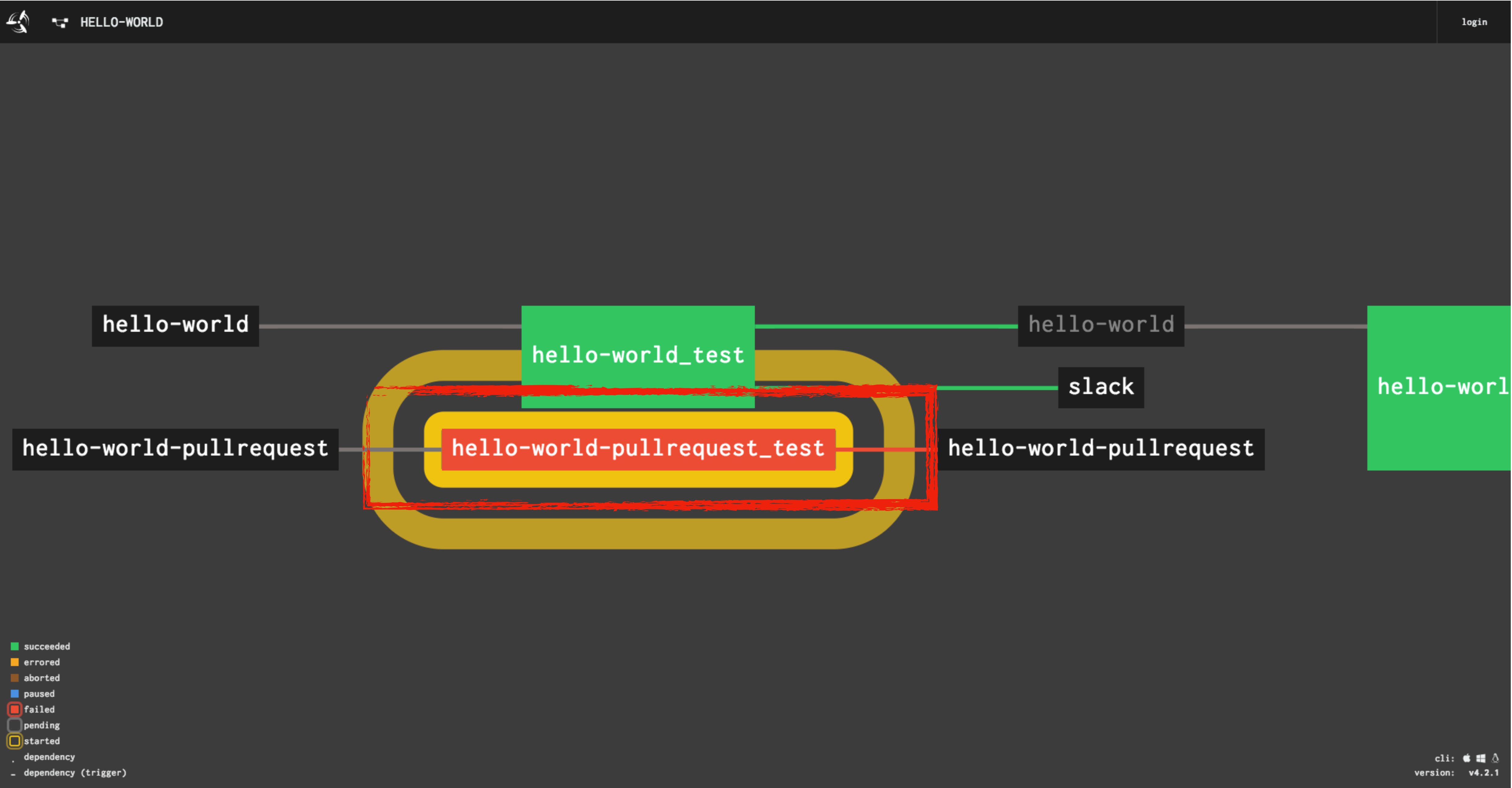
48

```
platform: linux
image_resource:
 type: docker-image
 source: {repository: gradle, tag: 4.10.2-jdk8-alpine}
inputs:
- name: source
outputs:
- name: docker
caches:
- path: source/.gradle/caches
run:
 dir: source
 user: root
 path: sh
 args:
 - -exc
 - |
 gradle --gradle-user-home=.gradle build

 cp build/libs/source.jar ../docker/application.jar
 cp ci/Dockerfile > ../docker/Dockerfile

 apk add --no-cache --no-progress git
 git describe --tags | sed 's/^v//' > ../docker/version
```

# Concourse CI v4.x.x - Job "\*-pullrequest\_test"



# Concourse CI v4.x.x - Job "\*-pullrequest\_test"

50

 HELLO-WORLD / <> hello-world-pullrequest\_test login

hello-world-pullrequest\_test #8

started 17m 53s ago  
finished 17m 0s ago  
duration 53s

14 13 12 11 10 9 8 7 6 5 4 3 2 1

↓ hello-world-pullrequest

ref 863c96766fa5a2c9472667079131fb5b894fa4e3 ✓

>\_ unit ✓

>\_ checkstyle ✓

↑ hello-world-pullrequest

↓ hello-world-pullrequest

↑ hello-world-pullrequest

ver static ✓

↓ hello-world-pullrequest

ver static ✓

```
jobs:
- name: hello-world-pullrequest_test
 plan:
 - get: hello-world-pullrequest
 trigger: true
 - aggregate:
 - task: unit
 file: hello-world-pullrequest/ci/unit.yml
 input_mapping: {source: hello-world-pullrequest}
 - task: checkstyle
 file: hello-world-pullrequest/ci/checkstyle.yml
 input_mapping: {source: hello-world-pullrequest}
 on_failure:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequestt, status: failure}
 on_success:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequest, status: success}
```

# Concourse CI v4.x.x - Job "\*-pullrequest\_test"

51

 HELLO-WORLD / <> hello-world-pullrequest\_test login

hello-world-pullrequest\_test #8

started 17m 53s ago  
finished 17m 0s ago  
duration 53s

14 13 12 11 10 9 8 7 6 5 4 3 2 1

↓ hello-world-pullrequest

ref 863c96766fa5a2c9472667079131fb5b894fa4e3 ✓

>\_ unit ✓

>\_ checkstyle ✓

↑ hello-world-pullrequest

↓ hello-world-pullrequest

↑ hello-world-pullrequest

ver static ✓

↓ hello-world-pullrequest

ver static ✓

```
jobs:
- name: hello-world-pullrequest_test
 plan:
 - get: hello-world-pullrequest
 trigger: true
 - aggregate:
 - task: unit
 file: hello-world-pullrequest/ci/unit.yml
 input_mapping: {source: hello-world-pullrequest}
 - task: checkstyle
 file: hello-world-pullrequest/ci/checkstyle.yml
 input_mapping: {source: hello-world-pullrequest}
 on_failure:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequestt, status: failure}
 on_success:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequest, status: success}
```

# Concourse CI v4.x.x - Job "\*-pullrequest\_test"

52

 HELLO-WORLD / <> hello-world-pullrequest\_test login

hello-world-pullrequest\_test #8

started 17m 53s ago  
finished 17m 0s ago  
duration 53s

14 13 12 11 10 9 8 7 6 5 4 3 2 1

↓ hello-world-pullrequest

ref 863c96766fa5a2c9472667079131fb5b894fae3 ✓

>\_ unit ✓

>\_ checkstyle ✓

↑ hello-world-pullrequest

↓ hello-world-pullrequest

↑ hello-world-pullrequest

ver static ✓

↓ hello-world-pullrequest

ver static ✓

```
jobs:
- name: hello-world-pullrequest_test
 plan:
 - get: hello-world-pullrequest
 trigger: true
 - aggregate:
 - task: unit
 file: hello-world-pullrequest/ci/unit.yml
 input_mapping: {source: hello-world-pullrequest}
 - task: checkstyle
 file: hello-world-pullrequest/ci/checkstyle.yml
 input_mapping: {source: hello-world-pullrequest}
 on_failure:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequestt, status: failure}
 on_success:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequest, status: success}
```

# Concourse CI v4.x.x - Job "\*-pullrequest\_test"

53

 HELLO-WORLD / <> hello-world-pullrequest\_test login

hello-world-pullrequest\_test #8

started 17m 53s ago  
finished 17m 0s ago  
duration 53s

14 13 12 11 10 9 8 7 6 5 4 3 2 1

↓ hello-world-pullrequest

ref 863c96766fa5a2c9472667079131fb5b894fae3 ✓

>\_ unit ✓

>\_ checkstyle ✓

↑ hello-world-pullrequest

↓ hello-world-pullrequest

↑ hello-world-pullrequest

ver static ✓

↓ hello-world-pullrequest

ver static ✓

```
jobs:
- name: hello-world-pullrequest_test
 plan:
 - get: hello-world-pullrequest
 trigger: true
 - aggregate:
 - task: unit
 file: hello-world-pullrequest/ci/unit.yml
 input_mapping: {source: hello-world-pullrequest}
 - task: checkstyle
 file: hello-world-pullrequest/ci/checkstyle.yml
 input_mapping: {source: hello-world-pullrequest}
 on_failure:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequestt, status: failure}
 on_success:
 put: hello-world-pullrequest
 params: {path: hello-world-pullrequest, status: success}
```

# Преимущества!

54

- YAML - удобно, структурировано
- CLI для управления CI`кой
- Интеграция с чем угодно через resource
- Просто написать свой resource
- Job одновременно выполняет множество task`ов в связке с разными resource`ами
- Переиспользуемые файлы task`ов

# Недостатки?

- Требовательность к ресурсам
- Бывает, worker залипает
- Сложно повторно запустить job на устаревшем ресурсе
- Нет разделения доступа по ролям в одной team`е

# Вопросы?



## Ссылки:

- <https://concourse-ci.org>
- <https://github.com/devinotelecom>
- <https://hub.docker.com/u/devinotelecom>
- <https://t.me/magnaz>

