RANCHER & CONTINUOUS DELIVERY

DockerGrunn #6

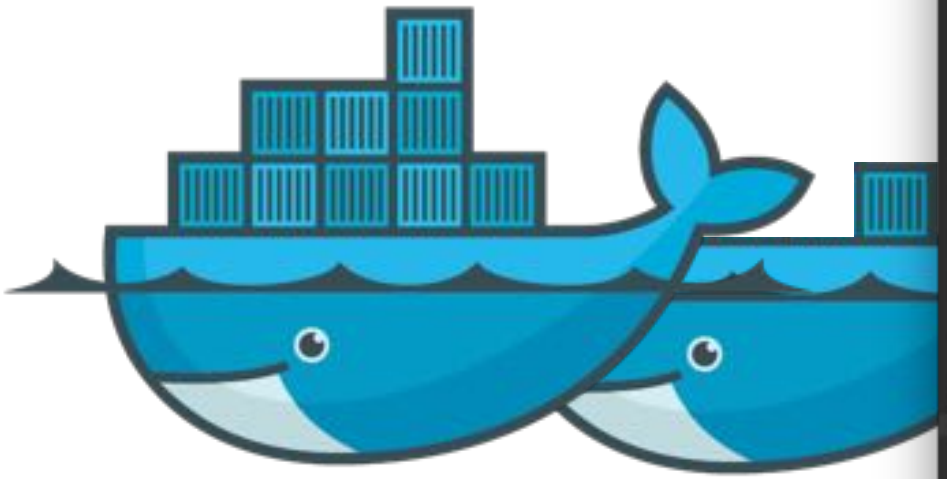
Johan van der Geest
Edwin Harmsma

TNO innovation for life

INTERESTING DEVELOPMENTS

+

DOCKER & DOCKER COMPOSE



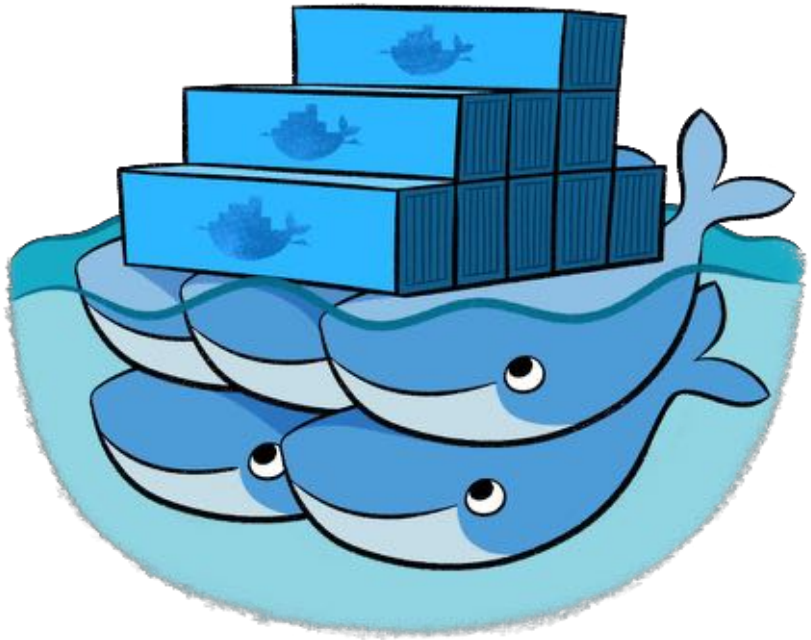
docker-compose.yml

```
rabbitmq:
  image: rabbitmq:3-management
  ports:
    - 5672:5672
    - 15672:15672
navigator:
  image: dockerhub.tno.nl/docker-demo/navigator:develop
  command: app --verbose
  environment: NAV_AMQP=amqp://rabbitmq:5672
  ports:
    - 8080:8080
  links:
    - rabbitmq
sentilyser:
  image: dockerhub.tno.nl/docker-demo/sentilyser:develop
  environment:
    - SA_PUBLISH_BROKER=rabbitmq
    - LANG=en_US.UTF-8
  restart: always
  links:
    - rabbitmq
```

Line 22, Column 1 0 misspelled words Spaces: 2 YAML

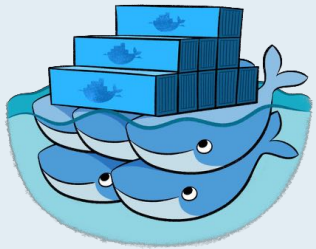
IT RUNS FINE ON A SINGLE MACHINE…

BUT WHAT ABOUT MORE THAN ONE WHALE?



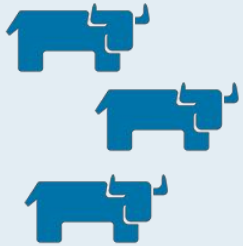
RANCHER

DOCKER CLUSTERING



Swarm

Kubernetes



Cattle

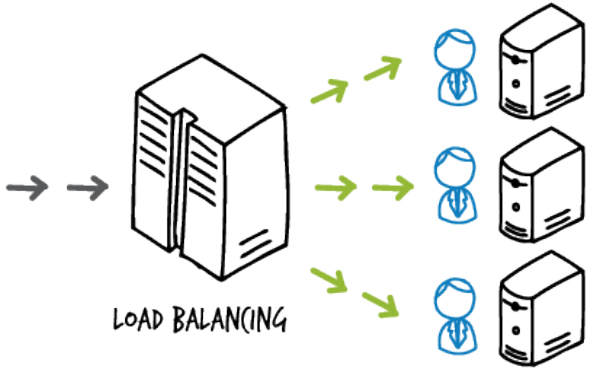
Mesos

Marathon

RANCHER FEATURES

Authentication
Environments
Audit trails

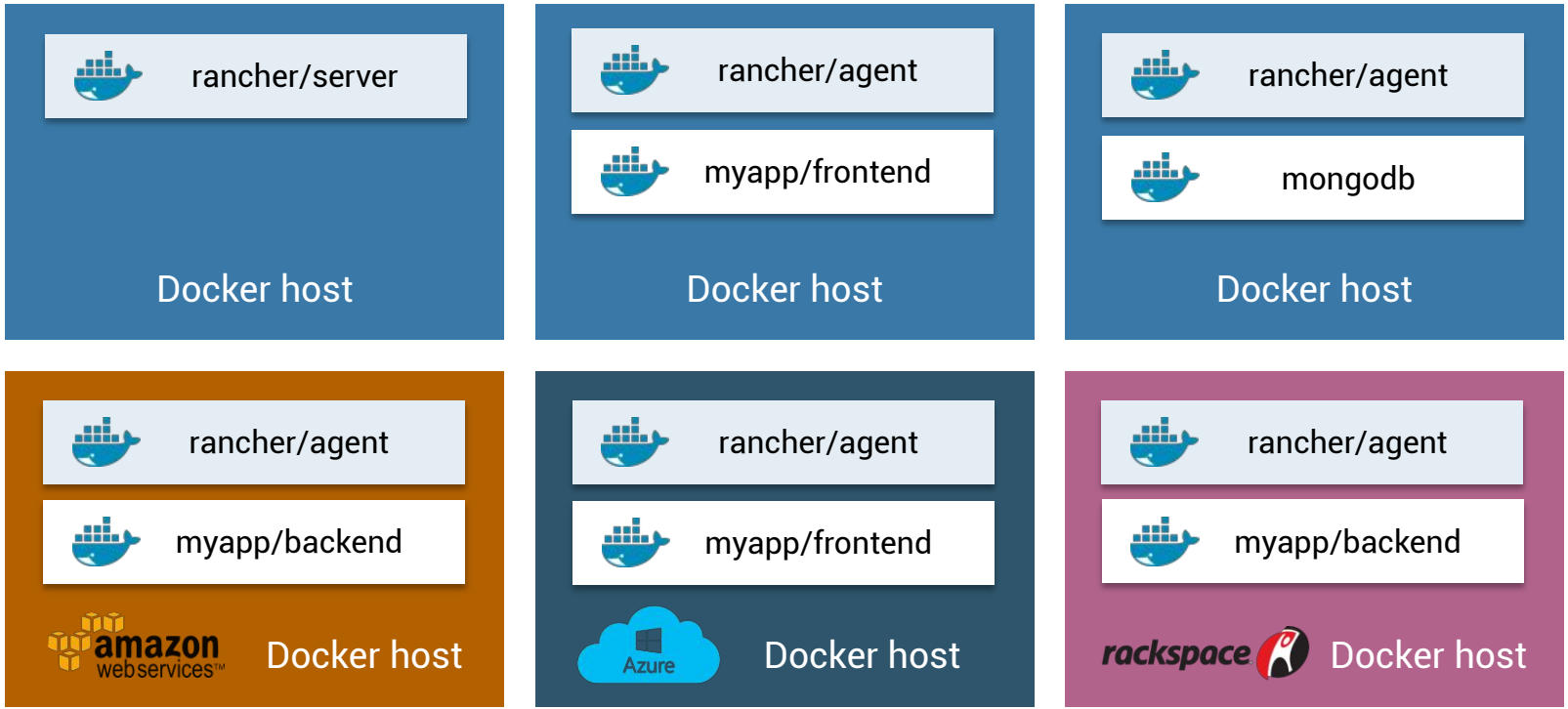
Service catalog



(Auto)scaling

Storage pools
Distributed storage (Convoy)

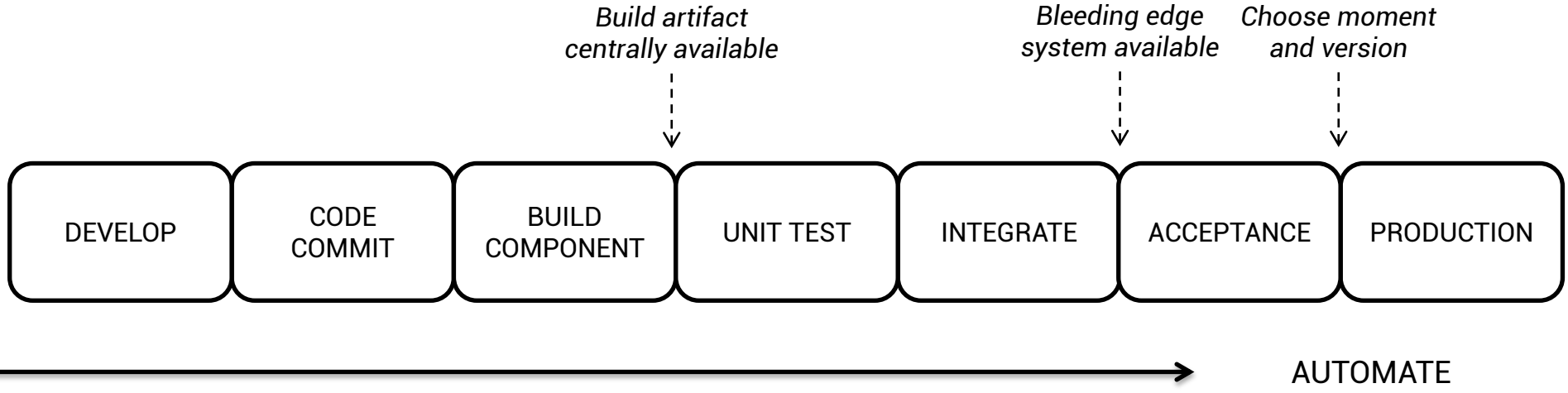
RANCHER SERVER & AGENTS



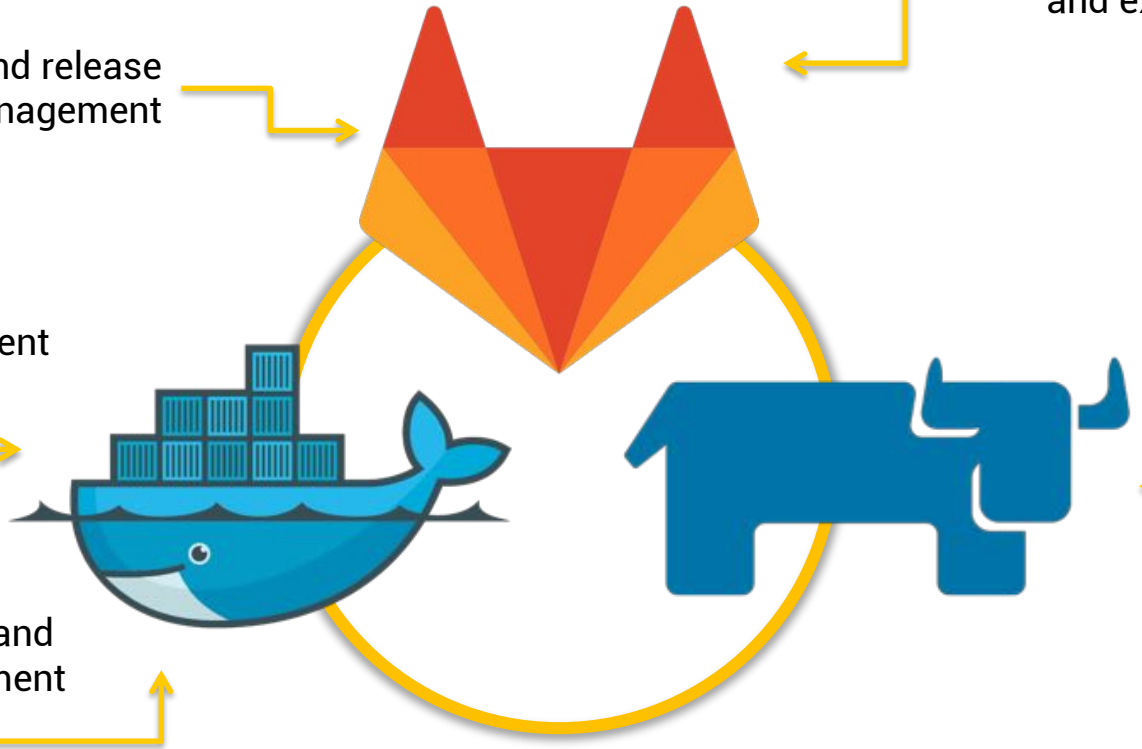

DEMO

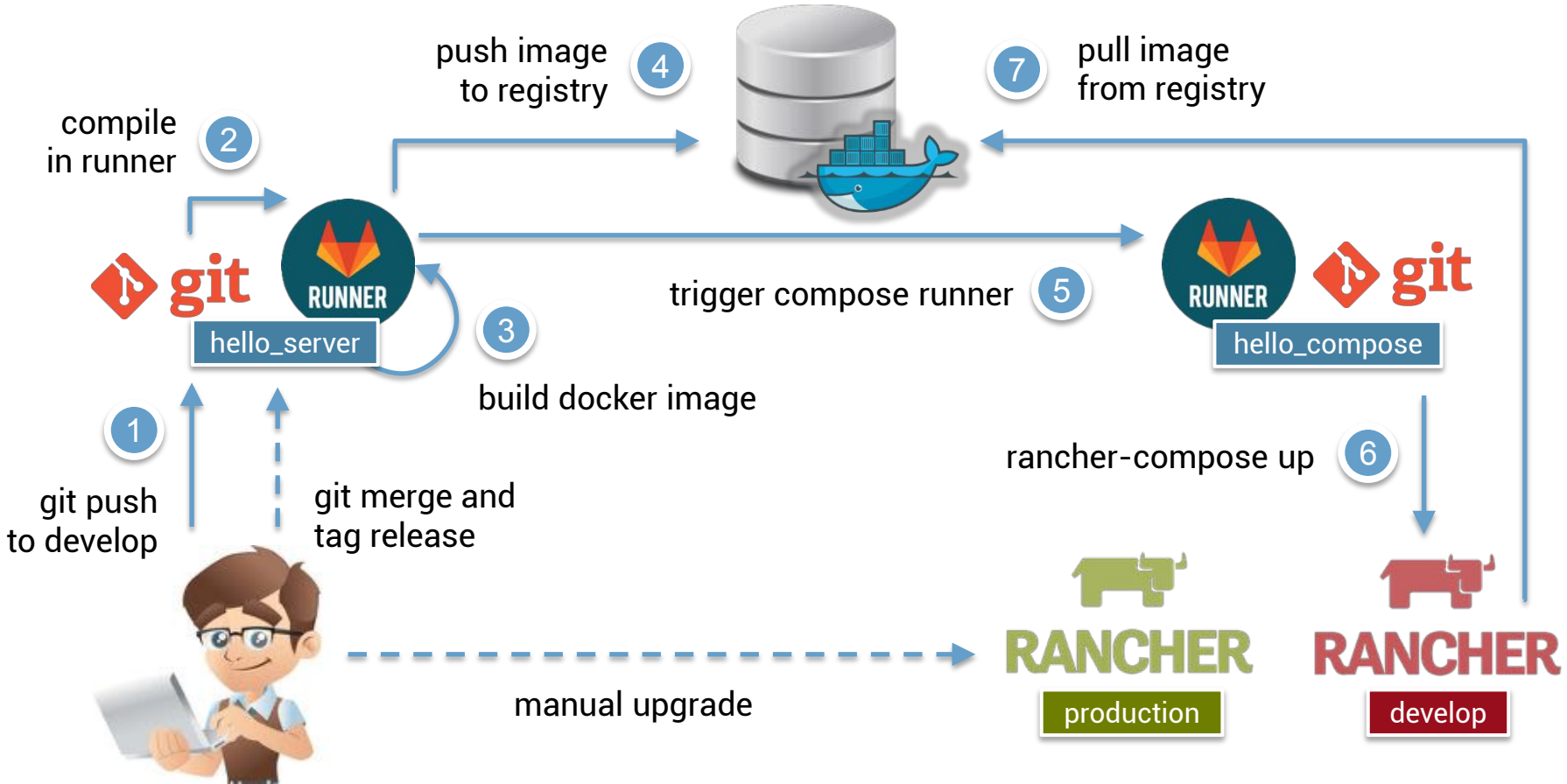

TNO innovation for life
CONTINUOUS DELIVERY
& RANCHER

CONTINUOUS DELIVERY

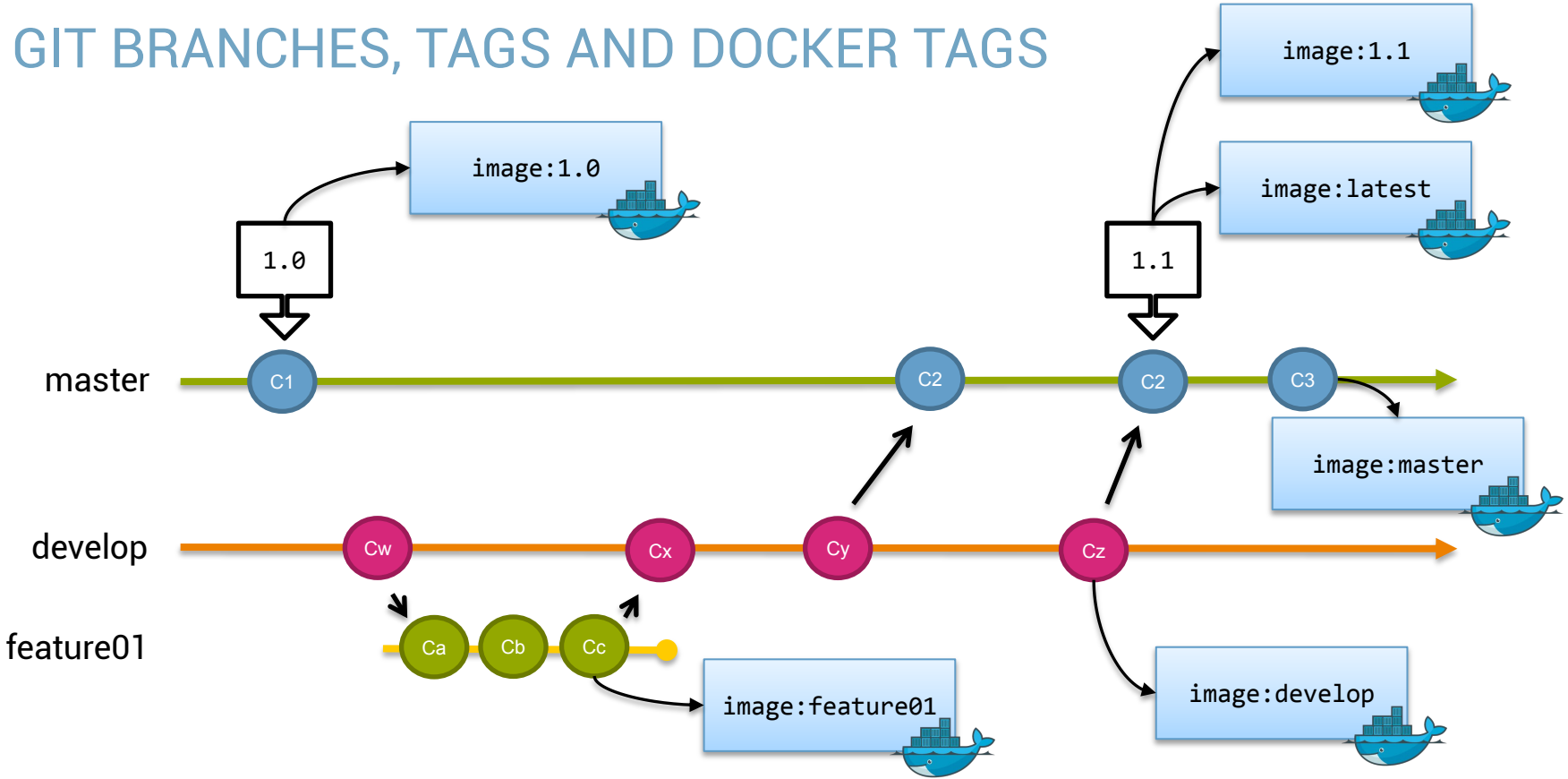


CONTINUOUS DELIVERY STACK

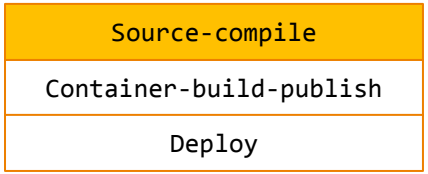



push image
to registry
4
7
pull image
from registry
compile
in runner
2
git
RUNNER
hello_server
3
build docker image
trigger compose runner
5
RUNNER
git
hello_compose
1
git push
to develop
git merge and
tag release
rancher-compose up
6
RANCHER
production
RANCHER
develop
manual upgrade

GIT BRANCHES, TAGS AND DOCKER TAGS



SOURCE COMPILE JOB




```
cargo-build:
  stage: source-compile
  image: hub.servicelab.org/rust/rustup
  script:
    - cargo build --release --target x86_64-unknown-linux-musl
  artifacts:
    paths:
      - target/x86_64-unknown-linux-musl/release/hello_server
  tags:
    - docker
  cache:
    key: cargo-cache
    paths:
      - .cargo
      - target/
```

Docker image used to setup build environment

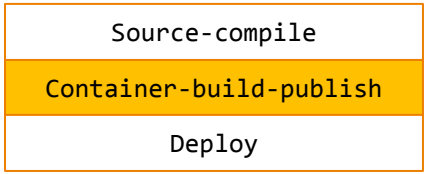
Command executed inside container

Preserve compiled binary after build

Select appropriate runner that can run Docker containers

Shared cache between all builds of this project

BUILD AND PUBLISH CONTAINER JOB




```
build-and-publish-to-docker:
  stage: container-build-publish
  script:
    - >
      docker build
      --build-arg CI_BUILD_ID=${CI_BUILD_ID}
      --build-arg CI_BUILD_GIT_REV=${CI_BUILD_REF}
      --pull --no-cache
      -t ${IMAGE_NAME} .
    - docker images ${IMAGE_NAME}
    - docker push ${IMAGE_NAME}
    - docker rmi ${IMAGE_NAME}
  tags:
    - shell
```

AUTO DEPLOY TO RANCHER

Source-compile

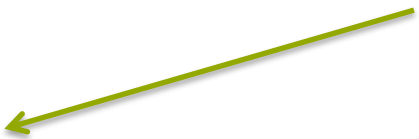
Container-build-publish

Deploy

```
image: monostream/rancher-compose

before_script:
  - source rancher.env

upgrade-service:
  script:
    - >
      rancher-compose
      -p hello-develop
      -f rancher-compose.develop.yml
      up
      -d
      --confirm-upgrade
      --force-upgrade
      --pull
      --upgrade ${service}
```



Trigger parameter passed by `hello_server` or `hello_client` projects


DEMO


THANKS FOR
YOUR ATTENTION
WE'RE
HIRING!
tno.nl/careers

BACKUP SLIDES

JOBS AND STAGES

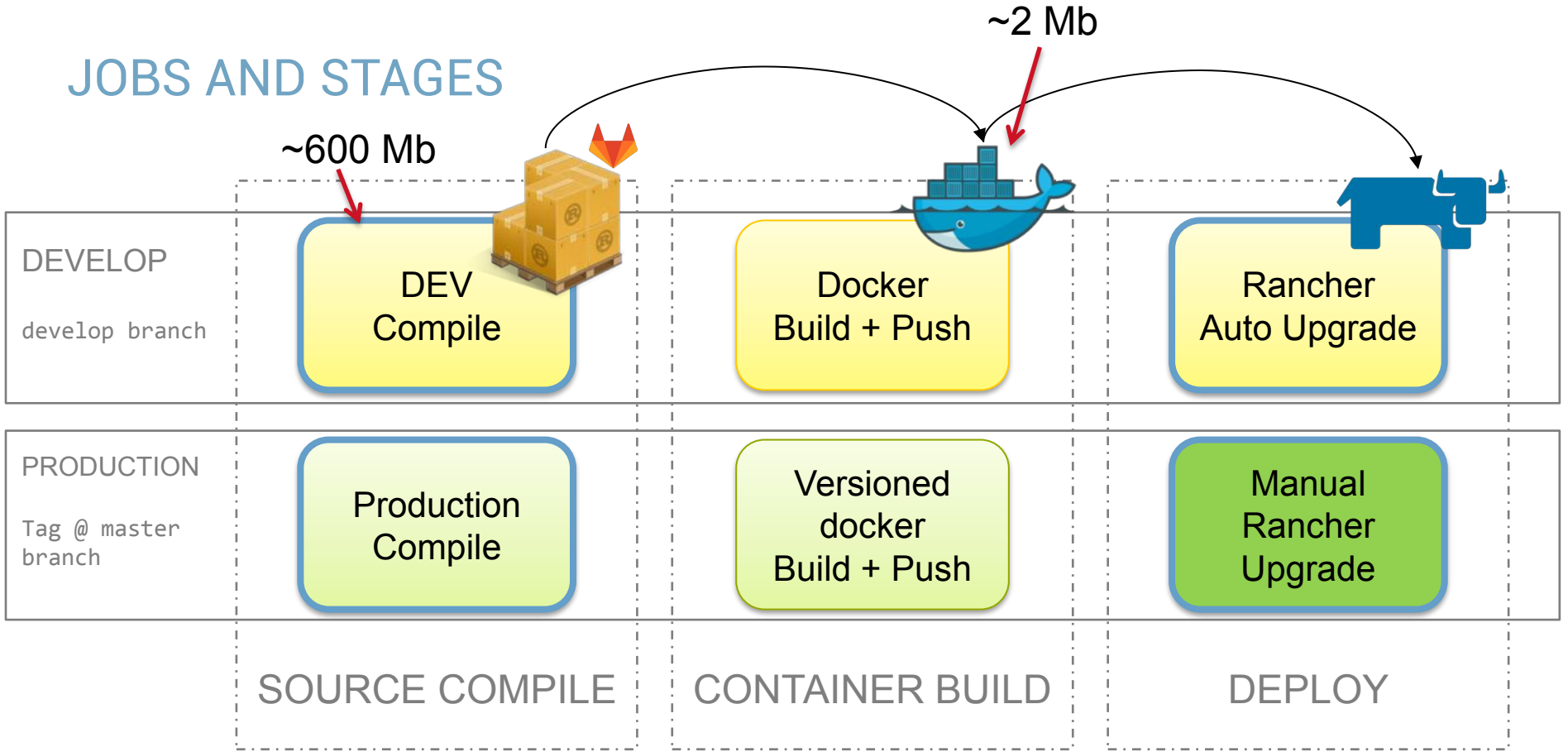
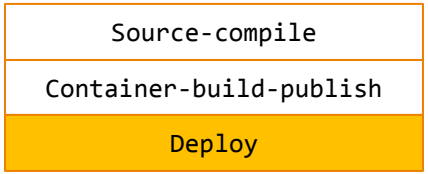


IMAGE SIZE: COMPILE VS DEPLOY CONTAINER

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hub.servicelab.org/docker-demo/hello_server	latest	deb921e8a18e	28 hours ago	2.089 MB
hub.servicelab.org/rust/rustup	latest	0920bbe2377d	5 days ago	576.3 MB
debian	latest	bb5d89f9b6cb	3 weeks ago	125.1 MB

PRODUCTION RELEASE: image:latest




```
production-release:
  stage: deploy
  script:
    - git branch --contains=`git show-ref --tags -s ${CI_BUILD_TAG}` master
    - docker pull ${IMAGE_NAME}
    - docker tag ${IMAGE_NAME} ${REGISTRY_URL}/hello_server:latest
    - docker push ${REGISTRY_URL}/hello_server:latest
    - docker rmi ${REGISTRY_URL}/hello_server:latest
  tags:
    - shell
  only:
    - /^([0-9]+\.)+([0-9]+)$/
  except:
    - branches
```