OSW2021
OPENSOURCEWEEK

GITLAB SU CLOUD IBRIDO: DALLA METODOLOGIA DEVOPS A QUELLA DEVSECOPS

Relatore
Ignazio Pedone, BE

BinarioEtico

#OSW2021

RIOS RETE ITALIANA OPEN SOURCE

How to build a DevSecOps pipeline?

- clear **strategy** and **design**
- **Source Code Management (SCM)** tools
- **CI/CD frameworks** for pipeline definition and execution
- private, public or hybrid **cloud platforms**

Gitlab for CI/CD pipelines

- Gitlab is one of the most advanced framework for:
  - SCM (Git), issue tracking, …
  - CI/CD automation (pipelines)

- the **Dev(Sec)Ops** platform

- **jobs** in Gitlab's domain are a set of tasks
  - associated to atomic functions within the pipeline (e.g., Static Code Analysis)
  - executed in parallel on different environments
  - triggered at a specific time (or "stage")

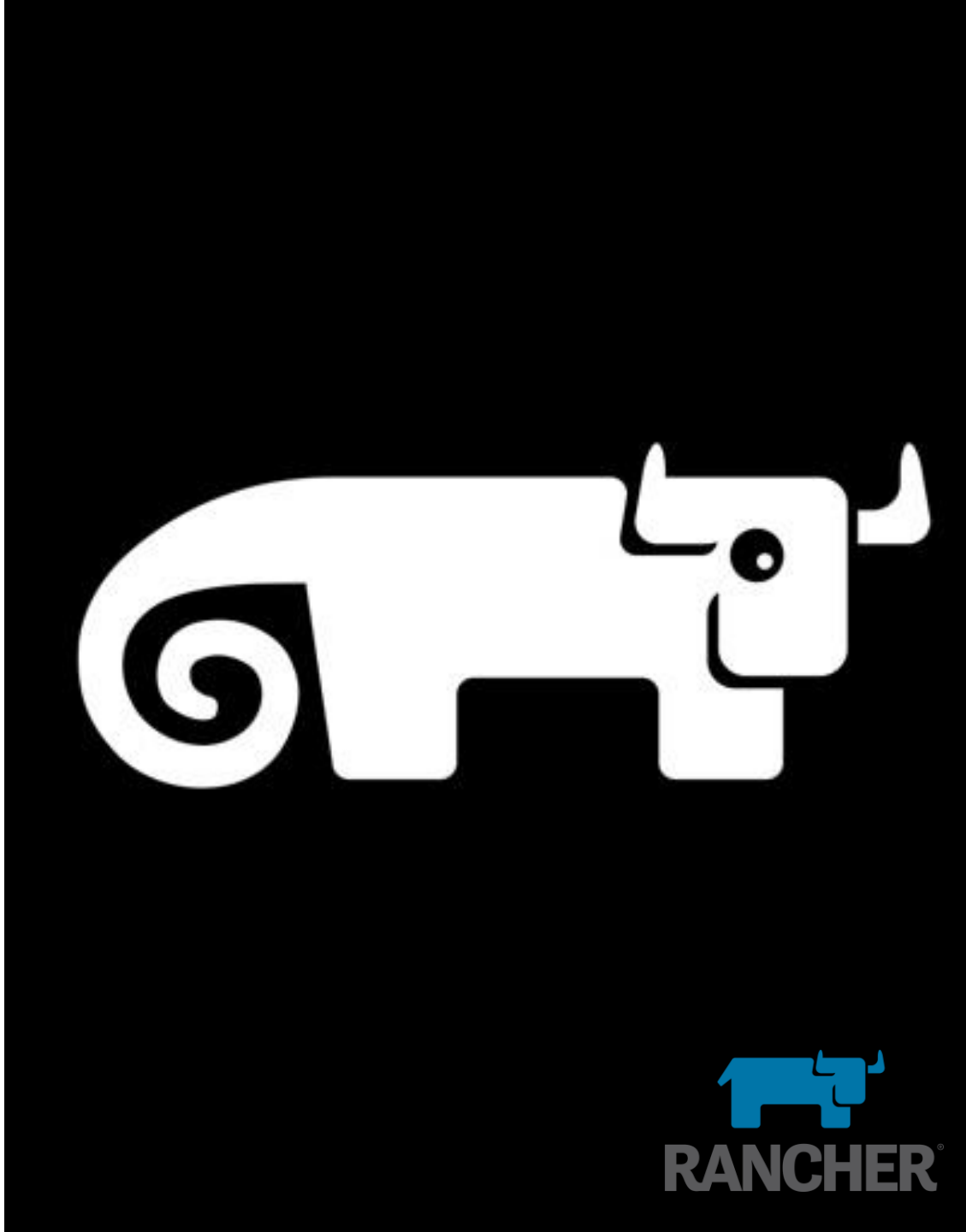
Gitlab for CI/CD pipelines II

- **stages** are the “steps” within the CI/CD pipeline
  - a job could be associated to a certain stage
  - within a stage all the related jobs are executed in parallel

- **runners** are the environments in which jobs run
  - could be VMs, physical nodes, Docker containers
  - no need for services to be exposed (a runner could be installed anywhere)

Rancher 101

- container **orchestration platform**
  - deploy and manage the lifecycle of containers
  - thousands of microservices
- several **container runtimes** supported
  - Docker, containerd, cri-o
- built on top of **Kubernetes (k8s)** to:
  - enhance its security features
  - extend its templating capabilities (i.e., using Helm)
  - manage multiple clusters and projects



Let’s be practical: a use case

- CI/CD pipeline composed of three stages:
  - **Test, Build, Deploy**

- **Node.js web application**
  - source code is managed by an on-premises Gitlab instance

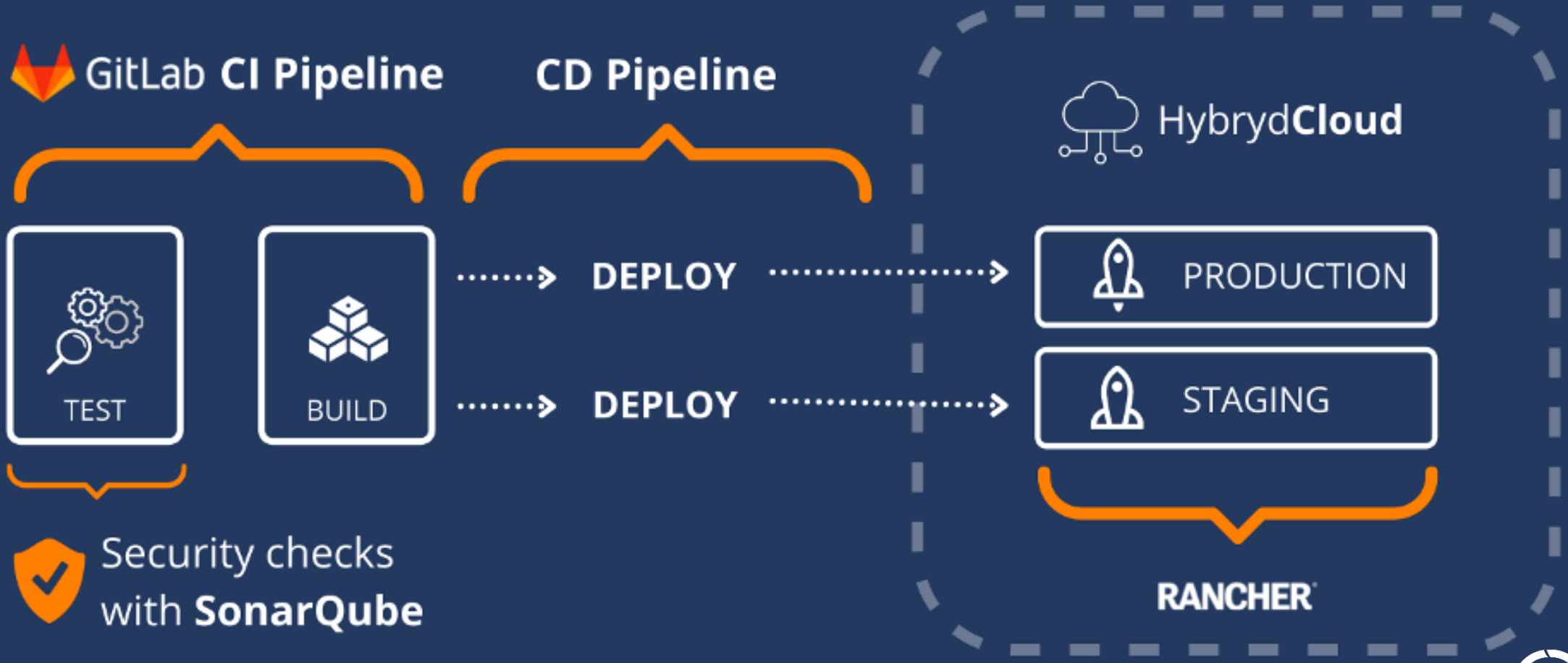
- Static Code Analysis with SonarQube **(Test)**
  - SonarQube installed on-premises
  - sonar-scanner deployed as an ephemeral Docker-based Gitlab runner

Let’s be practical: a use case II

- Automatic build of a Docker image and push on a specific registry **(Build)**
  - build process runs on a specific runner
  - push on DockerHub (could be any registry)

- Deploy on one of two environments based on the type of branch **(Deploy)**
  - deployment of the service on two different Rancher projects: development, production
  - based on the repository branch: either **dev** or **master**
  - could be deployed on heterogeneous environment (hybrid solutions)
    - AWS + Rancher, Rancher + OpenShift

From **DevOps** to **DevSecOps**



OSW2021
OPENSOURCEWEEK

Let’s be practical: gitlab-ci.yml

```yaml
stages:
  - analyze
  - docker-build
  - deploy


analyze:
  stage: analyze
  image:
    name: sonarsource/sonar-scanner-cli:4.5
    entrypoint: [""]
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"
    GIT_DEPTH: 0
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  script:
    - sonar-scanner
```

Stage definition

Test (SCA)

Let’s be practical: gitlab-ci.yml II

RETE ITALIANA OPEN SOURCE

Build

```
docker-image:
  stage: docker-build
  before_script:
    - docker info
    - docker login -u $DOCKERHUB_USER -p $DOCKERHUB_PASSWORD
  script:
    - docker build -f $CI_PROJECT_DIR/build/Dockerfile -t binarioetico/test:$CI_COMMIT_SHORT_SHA .
    - docker push binarioetico/test:$CI_COMMIT_SHORT_SHA
    - docker image rm binarioetico/test:$CI_COMMIT_SHORT_SHA
```

OSW2021 OPENSOURCEWEEK

Let’s be practical: gitlab-ci.yml III

RETE ITALIANA OPEN SOURCE

Deploy to dev environment

```
deploy_to_dev:
  stage: deploy
  image: badouralix/rancher-cli
  before_script:
    - export RANCHER_SECRET_KEY=$(echo $RANCHER_SECRET_KEY_DEV)
    - export RANCHER_ACCESS_KEY=$(echo $RANCHER_ACCESS_KEY_DEV)
  script:
    - apk add curl
    - cd /usr/bin; curl -LO "https://dl.k8s.io/release/$(curl -L
    - echo "192.168.1.198 myrancher.be.it" >> /etc/hosts
    - rancher login "$RANCHER_URL" -t "$RANCHER_BEARER_TOKEN" --c
    - sed -i "s/\$TAG/$CI_COMMIT_SHORT_SHA/g" $CI_PROJECT_DIR/dep
    - rancher kubectl apply -f $CI_PROJECT_DIR/deployment.yaml --
  rules:
    - if: '$CI_COMMIT_BRANCH == "dev"'
      when: on_success
    - when: never
```

Deploy to prod environment

```
deploy_to_prod:
  stage: deploy
  image: badouralix/rancher-cli
  before_script:
    - export RANCHER_SECRET_KEY=$(echo $RANCHER_SECRET_KEY_DEV)
    - export RANCHER_ACCESS_KEY=$(echo $RANCHER_ACCESS_KEY_DEV)
  script:
    - apk add curl
    - cd /usr/bin; curl -LO "https://dl.k8s.io/release/$(curl -L -
    - echo "192.168.1.198 myrancher.be.it" >> /etc/hosts
    - rancher login "$RANCHER_URL" -t "$RANCHER_BEARER_TOKEN" --co
    - sed -i "s/\$TAG/$CI_COMMIT_SHORT_SHA/g" $CI_PROJECT_DIR/depl
    - rancher kubectl apply -f $CI_PROJECT_DIR/deployment.yaml --r
  rules:
    - if: '$CI_COMMIT_BRANCH == "master"'
      when: on_success
    - when: never
```

OSW2021 OPENSOURCEWEEK

Grazie!