REFERENSI CEPAT BITBUCKET PIPELINES

CI/CD pipeline, caching, artifact, deployment

Dasar Pipeline

Cara Kerja

`bitbucket-pipelines.yml`	File konfigurasi di root repo
`Docker containers`	Setiap step berjalan di container tersendiri
`Trigger`	Push, PR, tag, jadwal, atau manual
`Build minutes`	Kuota bergantung pada paket berlangganan

Mengaktifkan Pipeline

```
# Repository Settings → Pipelines → Enable
# Add bitbucket-pipelines.yml to repo root
# First push triggers the pipeline
```

bitbucket-pipelines.yml

Konfigurasi Minimal

```
image: node:20
pipelines:
  default:
    - step:
        script:
          - npm install
          - npm test
```

Pipeline Per Branch

```
pipelines:
  branches:
    main:
      - step:
          script:
            - npm run build
            - npm run deploy
```

Pipeline Tag & Pull Request

```
pipelines:
  tags:
    'v*':
      - step:
          script:
            - npm run release
  pull-requests:
    '**':
      - step:
          script:
            - npm test
```

Step

Opsi Step

`name`	Nama tampilan step
`image`	Override Docker image global
`script`	Daftar perintah shell yang dijalankan
`size`	Memori 1x (4GB) atau 2x (8GB)
`max-time`	Timeout dalam menit (default 120)
`trigger`	`manual` untuk step yang hanya bisa dipicu manual

Step Paralel

```
- parallel:
    - step:
        name: "Lint"
        script:
          - npm run lint
    - step:
        name: "Test"
        script:
          - npm test
```

Step Manual

```
- step:
    name: "Deploy to Production"
    trigger: manual
    script:
      - ./deploy.sh prod
```

Variabel

Jenis Variabel

`Repository variables`	Settings → Pipelines → Variables
`Deployment variables`	Dicakup ke environment deployment tertentu
`Secured variables`	Terenkripsi, disembunyikan di log
`Pipeline variables`	Didefinisikan inline di YAML

Menggunakan Variabel

```
pipelines:
  default:
    - step:
        script:
          - echo $MY_VAR
          - docker login -u $DOCKER_USER -p $DOCKER_PASS
```

Variabel Bawaan

`$BITBUCKET_COMMIT`	SHA commit lengkap
`$BITBUCKET_BRANCH`	Nama branch
`$BITBUCKET_TAG`	Nama tag (pipeline tag)
`$BITBUCKET_BUILD_NUMBER`	Nomor build yang terus bertambah
`$BITBUCKET_REPO_SLUG`	Slug repository

Caching

Cache Bawaan

```
- step:
    caches:
      - node        # ~/.npm
      - pip         # ~/.cache/pip
      - docker      # Docker layer cache
    script:
      - npm install
      - npm test
```

Cache Kustom

```
definitions:
  caches:
    gradle: ~/.gradle/caches
    mylibs: vendor/libs
pipelines:
  default:
    - step:
        caches:
          - gradle
        script:
          - ./gradlew build
```

Perilaku Cache

`Durasi`	Cache kedaluwarsa setelah 7 hari
`Cakupan`	Dibagikan ke semua pipeline dalam repo
`Hapus`	Pipelines → Caches → Delete

Artifact

Mengirim File Antar Step

```
- step:
    name: "Build"
    script:
      - npm run build
    artifacts:
      - dist/**
- step:
    name: "Deploy"
    script:
      - ls dist/    # artifacts available
      - ./deploy.sh
```

Opsi Artifact

`artifacts`	Pola glob untuk file yang diteruskan
`Download`	Tersedia di step berikutnya secara otomatis
`Ukuran maks`	1 GB per step
`Retensi`	Tersedia 14 hari setelah build

Deployment

Environment Deployment

```
- step:
    name: "Deploy Staging"
    deployment: staging
    script:
      - ./deploy.sh staging
- step:
    name: "Deploy Production"
    deployment: production
    trigger: manual
    script:
      - ./deploy.sh prod
```

Jenis Environment

`test`	Environment pengujian
`staging`	Environment pra-produksi
`production`	Lingkungan live, terlacak di dashboard

Pola Umum

Docker Build & Push

```
- step:
    services:
      - docker
    script:
      - docker build -t myapp:$BITBUCKET_COMMIT .
      - docker login -u $DOCKER_USER -p $DOCKER_PASS
      - docker push myapp:$BITBUCKET_COMMIT
```

Service Container

```
definitions:
  services:
    postgres:
      image: postgres:16
      variables:
        POSTGRES_DB: testdb
        POSTGRES_PASSWORD: secret
pipelines:
  default:
    - step:
        services:
          - postgres
        script:
          - npm test
```

Step Kondisional dengan Pipe

```
- step:
    name: "Deploy to S3"
    script:
      - pipe: atlassian/aws-s3-deploy:1.1.0
        variables:
          AWS_ACCESS_KEY_ID: $AWS_KEY
          AWS_SECRET_ACCESS_KEY: $AWS_SECRET
          S3_BUCKET: my-bucket
          LOCAL_PATH: dist/
```