

BITBUCKET PIPELINES THAM KHẢO NHANH

CI/CD pipeline, caching, artifact, deployment

Cơ Bản Pipeline

Cách Hoạt Động
bitbucket-pipelines.yml File cấu hình ở thư mục gốc repo
Docker containers Mỗi step chạy trong container riêng
Triggers Push, PR, tag, lịch trình, hoặc thủ công
Build minutes Hạn ngạch phụ thuộc vào gói dịch vụ

Bật Pipelines

```
# Repository Settings → Pipelines → Enable  
# Thêm bitbucket-pipelines.yml vào thư mục gốc  
# Push đầu tiên sẽ kích hoạt pipeline
```

bitbucket-pipelines.yml

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

Pipeline Theo Branch

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

Pipeline Theo Tag & Pull Request

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

Steps

Tùy Chọn Step

name Tên hiển thị của step
image Ghi đè Docker image toàn cục
script Danh sách lệnh shell cần chạy
size 1x (4GB) hoặc 2x (8GB) bộ nhớ
max-time Timeout tính bằng phút (mặc định 120)
trigger manual cho step chỉ chạy thủ công

Step Song Song

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

Step Thủ Công

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

Biến

Các Loại Biến

Repository variables Settings → Pipelines → Variables
Deployment variables Giới hạn trong môi trường deployment
Secured variables Mã hóa, ẩn trong log
Pipeline variables Định nghĩa trực tiếp trong YAML

Sử Dụng Biến

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

Biến Built-in

BITBUCKET_COMMIT SHA commit đầy đủ
BITBUCKET_BRANCH Tên branch
BITBUCKET_TAG Tên tag (pipeline theo tag)
BITBUCKET_BUILD_NUMBER Số build tăng dần
BITBUCKET_REPO_SLUG Slug của repository

Caching

Cache Có Sẵn

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

Cache Tùy Chỉnh

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

Hành Vi Cache

Thời hạn Cache hết hạn sau 7 ngày
Phạm vi Chia sẻ trên tất cả pipeline trong repo
Xóa Pipelines → Caches → Delete

Artifacts

Truyền File Giữa Các Step

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

Tùy Chọn Artifact

artifacts Glob pattern cho file cần truyền
Download Tự động có sẵn ở các step tiếp theo
Max size 1 GB mỗi step
Retention Có sẵn trong 14 ngày sau khi build

Deployments

Môi Trường Deployment

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

Loại Môi Trường

test Môi trường kiểm thử
staging Môi trường tiền production
production Môi trường live, theo dõi trên dashboard

Các Pattern Thường Gặp

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 Containers

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

Step Điều Khiển với 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/
```