

Bitbucket Pipelines クイックリファレンス

CI/CD パイプライン、キャッシュ、アーティファクト、デプロイ

パイプラインの基本

仕組み

bitbucket-pipelines.yml	リポジトリルートの設定ファイル
Docker コンテナ	各ステップが独自のコンテナで実行
トリガー	プッシュ、PR、タグ、スケジュール、または手動
ビルド時間	クォータはプランによって異なる

パイプラインの有効化

```
# リポジトリ設定 → Pipelines → 有効化
# bitbucket-pipelines.yml をリポジトリルートに追加
# 最初のプッシュでパイプラインが起動
```

bitbucket-pipelines.yml

最小構成

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

ブランチ固有のパイプライン

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

タグとPRのパイプライン

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

ステップ

ステップオプション

name	ステップの表示名
image	グローバルの Docker イメージを上書き
script	実行するシェルコマンドのリスト
size	1x (4GB) または 2x (8GB) のメモリ
max-time	タイムアウト (分) デフォルト 120
trigger	manual で手動実行専用ステップ

並列ステップ

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

手動ステップ

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

変数

変数の種類

リポジトリ変数	設定 → Pipelines → Variables
デプロイ変数	デプロイ環境にスコープされる
セキュア変数	暗号化され、ログでマスク
パイプライン変数	YAML にインラインで定義

変数の使用

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

組み込み変数

\$BITBUCKET_COMMIT	完全なコミット SHA
\$BITBUCKET_BRANCH	ブランチ名
\$BITBUCKET_TAG	タグ名 (タグパイプライン)
\$BITBUCKET_BUILD_NUMBER	インクリメントするビルド番号
\$BITBUCKET_REPO_SLUG	リポジトリスラッグ

キャッシュ

定義済みキャッシュ

```
- step:
  caches:
    - node # ~/.npm
    - pip # ~/.cache/pip
    - docker # Docker レイヤーキャッシュ
  script:
    - npm install
    - npm test
```

カスタムキャッシュ

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

キャッシュの動作

有効期限	7 日後に期限切れ
スコープ	リポジトリ内のすべてのパイプラインで共有
クリア	Pipelines → Caches → Delete

アーティファクト

ステップ間でファイルを渡す

```
- step:
  name: "Build"
  script:
    - npm run build
  artifacts:
    - dist/**
- step:
  name: "Deploy"
  script:
    - ls dist/ # アーティファクトが利用可能
    - ./deploy.sh
```

アーティファクトオプション

artifacts	渡すファイルのグロブパターン
ダウンロード	後続のステップで自動的に利用可能
最大サイズ	ステップあたり 1GB
保持期間	ビルド後 14 日間利用可能

デプロイ

デプロイ環境

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

環境の種類

test	テスト環境
staging	ステージング環境
production	本番環境、ダッシュボードで追跡

よくあるパターン

Docker ビルドとプッシュ

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

サービスコンテナ

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

Bitbucket Pipelines クイックリファレンス

パイプを使った条件付きステップ

```
- 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/
```