

# Referência Rápida Bitbucket Pipelines

Pipelines CI/CD, cache, artefatos, deploys

## Básico de Pipelines

### Como Funciona

<b>bitbucket-pipelines.yml</b>	Arquivo de configuração na raiz do repositório
<b>Docker containers</b>	Cada step executa em seu próprio container
<b>Triggers</b>	Push, PR, tag, agendamento ou manual
<b>Build minutes</b>	Cota depende do plano contratado

### Habilitar Pipelines

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

## bitbucket-pipelines.yml

### Configuração Mínima

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

### Pipelines por Branch

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

### Pipelines de Tag e Pull Request

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

## Steps

### Opções de Step

<b>name</b>	Nome de exibição do step
<b>image</b>	Sobrescrever imagem Docker global
<b>script</b>	Lista de comandos shell a executar
<b>size</b>	1x (4GB) ou 2x (8GB) de memória
<b>max-time</b>	Timeout em minutos (padrão 120)
<b>trigger</b>	manual para steps somente manuais

### Steps em Paralelo

```
- 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
```

## Variáveis

### Tipos de Variáveis

<b>Repository variables</b>	Settings → Pipelines → Variables
<b>Deployment variables</b>	Com escopo para um ambiente de deploy
<b>Secured variables</b>	Criptografadas, mascaradas nos logs
<b>Pipeline variables</b>	Definidas inline no YAML

### Usando Variáveis

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

### Variáveis Embutidas

<b>\$BITBUCKET_COMMIT</b>	SHA completo do commit
<b>\$BITBUCKET_BRANCH</b>	Nome da branch
<b>\$BITBUCKET_TAG</b>	Nome da tag (pipelines de tag)
<b>\$BITBUCKET_BUILD_NUMBER</b>	Número de build incremental
<b>\$BITBUCKET_REPO_SLUG</b>	Slug do repositório

## Cache

### Caches Predefinidos

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

### Cache Personalizado

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

### Comportamento do Cache

<b>Duration</b>	Caches expiram após 7 dias
<b>Scope</b>	Compartilhado entre todos os pipelines do repositório
<b>Clear</b>	Pipelines → Caches → Delete

## Artefatos

### Passar Arquivos Entre Steps

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

### Opções de Artefatos

<b>artifacts</b>	Padrões glob para arquivos a passar
<b>Download</b>	Disponível em steps subsequentes automaticamente
<b>Max size</b>	1 GB por step
<b>Retention</b>	Disponível por 14 dias após o build

## Deploys

### Ambientes de Deploy

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

### Tipos de Ambiente

<b>test</b>	Ambiente de testes
<b>staging</b>	Ambiente de pré-produção
<b>production</b>	Ambiente de produção, rastreado no dashboard

## Padrões Comuns

### Build e Push Docker

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

### Containers de Serviço

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

# Referência Rápida Bitbucket Pipelines

---

## Step Condicional com 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/
```