# Tutorial de criação de pipelines no Openshift para Projetos Maven

Sumário

[Tutorial de criação de pipelines no Openshift para Projetos Maven 1](#_Toc529523405)

[Projeto 2](#_Toc529523406)

[Configurações de Maven e Nexus 2](#_Toc529523407)

[Dockerfile 2](#_Toc529523408)

[pom.xml 3](#_Toc529523409)

[application-local.properties 3](#_Toc529523410)

[application.properties para homologação 3](#_Toc529523411)

[Deploy 4](#_Toc529523412)

[Template 4](#_Toc529523413)

[Renomear arquivo YAML 4](#_Toc529523414)

[Preparação do YAML 4](#_Toc529523415)

[Preparação do arquivo deploy.sh 4](#_Toc529523416)

[Execução do deploy 5](#_Toc529523417)

[Executando o Deploy 5](#_Toc529523418)

[Salvando os arquivos no GIT 5](#_Toc529523419)

[Openshift 6](#_Toc529523420)

[Acessando o Openshift 6](#_Toc529523421)

[Configuração do Pipeline 6](#_Toc529523422)

[Configurando a integração do Openshift com o GIT 6](#_Toc529523423)

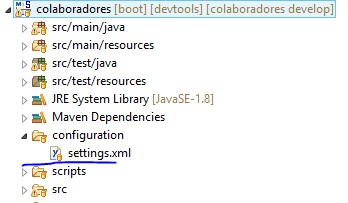
[Iniciando o Pipeline 8](#_Toc529523424)

## Projeto

Para criar um pipeline no Openshift o projeto Maven deverá seguir alguns padrões. A não utilização de um ou mais padrões poderá acarretar falha na execução do pipeline de homologação.

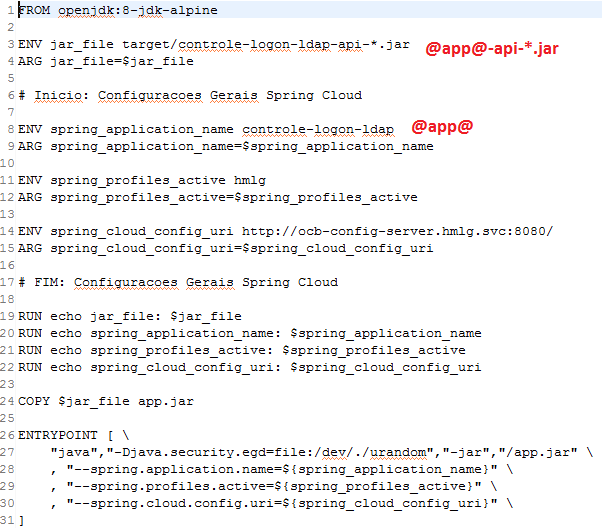
### Configurações de Maven e Nexus

Na raiz do projeto, deverá existir o diretório “configuration” com o arquivo “settings.xml” para que o Maven utilize as bibliotecas de projetos presentes no Nexus interno, preferencialmente.



### Dockerfile

O dockerfile deverá seguir as seguintes características:



## pom.xml

O arquivo pom.xml deverá conter a identificação do projeto:

Em groupId, o valor padrão atual é: “br.coop.somoscooperativismo”.

Em artifactId, o valor atual é o “@app@-api”.

O valor de version deverá ser atualizado a cada atualização de versão em produção do sistema.

<groupId>br.coop.somoscooperativismo</groupId>

<artifactId>controle-logon-ldap-api</artifactId>

<version>0.0.2-SNAPSHOT</version>

<packaging>jar</packaging>

### application-local.properties

No diretório “resources”, o arquivo “application.properties” deverá se chamar “application-local.properties” com as características do projeto para execução na máquina do desenvolvedor.

### application.properties para homologação

Para execução em homologação, o “application-local.properties” deverá ser copiado para o GIT, em <http://git.somoscooperativismo.coop.br/administradores/ocb-application-config> e renomeado para “@app@-hmlg.properties” e seguir as características do projeto para execução em homologação.

Exemplo de arquivos properties para homologação:

* *arrecadacao-processador-hmlg.properties*
* *juris-hmlg.properties*

Dando *push* deste arquivo para o diretório, o Openshift já adotará as propriedades salvas.

## Deploy

### Template

Baixar os arquivos de *template* do Maven em

<http://git.somoscooperativismo.coop.br/jenkins/OPENSHIFTORIGIN/tree/master/openshift/cicd/pipelines/templates/template-maven>

* *deploy.sh*
* *app-pipeline.yaml*

### Renomear arquivo YAML

Renomear o arquivo “app-pipeline.yaml” para (preferencialmente) “[spring\_application\_name]-pipeline.yaml”.

[spring\_application\_name] é uma variável encontrada no Dockerfile do projeto java.

Por exemplo:

* *controle-logon-ldap-pipeline.yaml*
* *juris-pipeline.yaml*

### Preparação do YAML

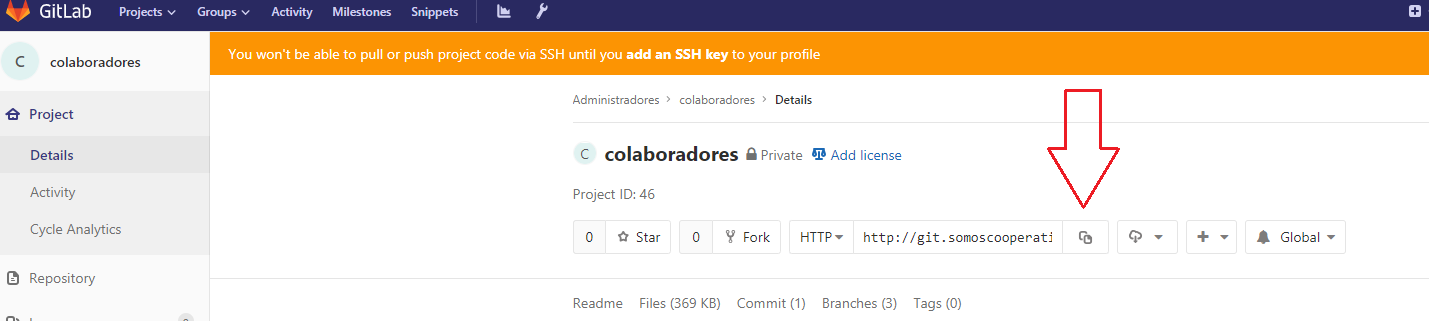
Substituir as variáveis **@app@** encontradas dentro do arquivo YAML pelo (preferencialmente) [spring\_application\_name].

Substituir a variável **@endereco@** encontrada dentro do arquivo YAML pelo endereço da aplicação a ser disponibilizado pelo Openshift.

Padrão do momento:

* *juris-front-****hmlg.cloudapps.ocb.org.br***
* *arrecadacao-processador-****hmlg.cloudapps.ocb.org.br***

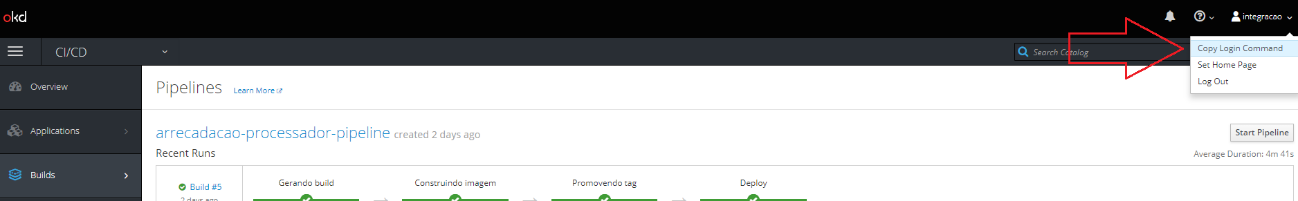
Substituir a variável **@git@** encontrada dentro do arquivo YAML pelo endereço do projeto no GIT.



O endereço pode ser copiado dentro do GIT.

### Preparação do arquivo deploy.sh

Substituir a variável **@app@** encontrada dentro do arquivo “deploy.sh” pelo (preferencialmente) [spring\_application\_name].



No arquivo, pode-se usar “oc login -u system:admin” ou copiar um *token* pelo portal.

### Execução do deploy

#### Copiando os arquivos para o SORENTO

É necessário copiar os arquivos preparados [yaml e sh] para o servidor Sorento.

Para copiar, utilize um aplicativo como o FileZilla, e conecte-se ao servidor com as seguintes configurações:

* **Host**: sorento.ocb.org.br
* **Usuário**: okd
* **Senha**: okd@123
* **Porta**: 22

Criar um diretório com o nome da aplicação (“controleLogon”, por exemplo) e transferir os arquivos modificados para o diretório.

### Executando o Deploy

Acessar o servidor Sorento utilizando uma aplicação como o “Putty.exe” utilizando as mesmas configurações de host/usuário do FileZilla.

Entrar no diretório criado com o nome da aplicação, onde estarão os arquivos “@app@-pipeline.yaml” e “deploy.sh”.

*Para garantir a execução dos arquivos, estamos atribuindo permissões especiais:*

*sudo chmod -x @app@-pipeline.yaml*

*sudo chmod -x deploy.sh*

Na sequência, executar o comando “sudo sh deploy.sh”. Este comando deverá criar o pipeline da aplicação Maven no Openshift.

### Salvando os arquivos no GIT

Dar um *push* com a pasta com o nome do sistema **@app@** e com os arquivos YAML e deploy.sh no GIT em:

<http://git.somoscooperativismo.coop.br/jenkins/OPENSHIFTORIGIN/tree/master/openshift/cicd/pipelines>

## Openshift

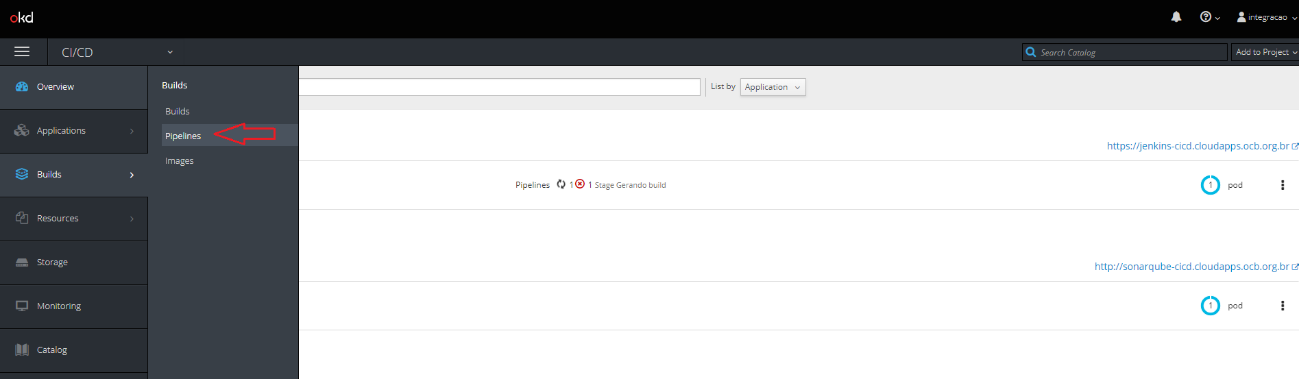
### Acessando o Openshift

Para acessar o Openshift, as configurações atuais são:

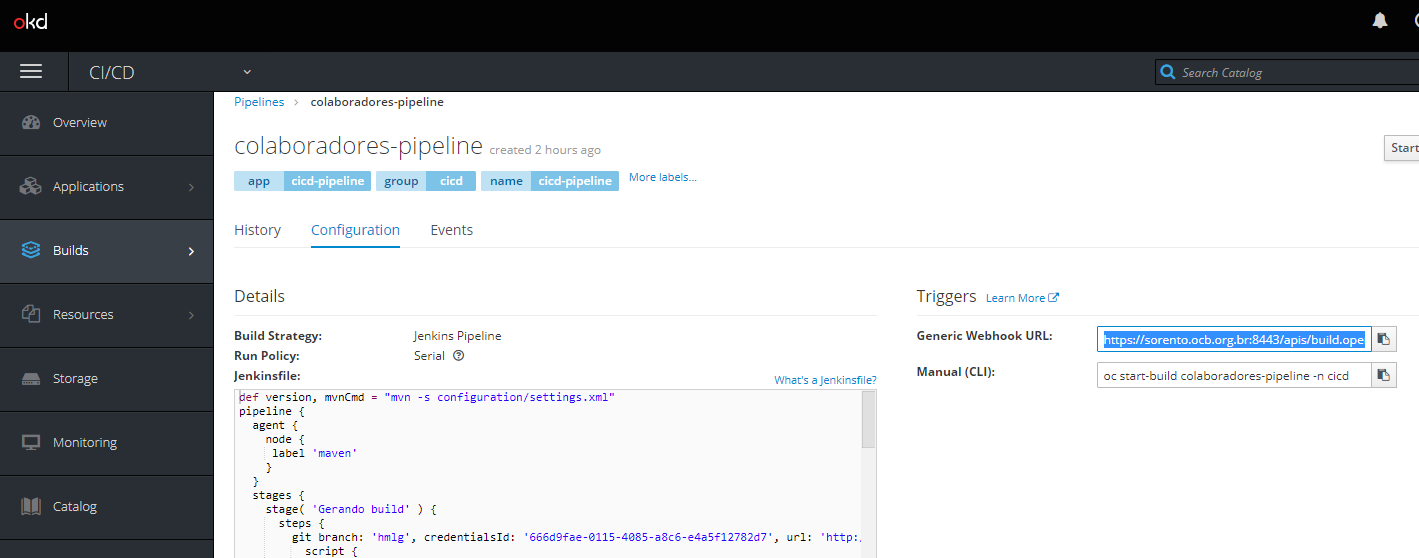
* **Endereço**: <https://sorento.ocb.org.br:8443/>
* **Login**: integracao
* **Senha**: ocb@123

### Configuração do Pipeline

Para configurar o pipeline criado, deve-se acessar o projeto CI/CD, no menu Builds/Pipelines.



Entrar no pipeline criado, na aba “Configuration” e copiar o Webhook URL para configurar a integração com o GIT.



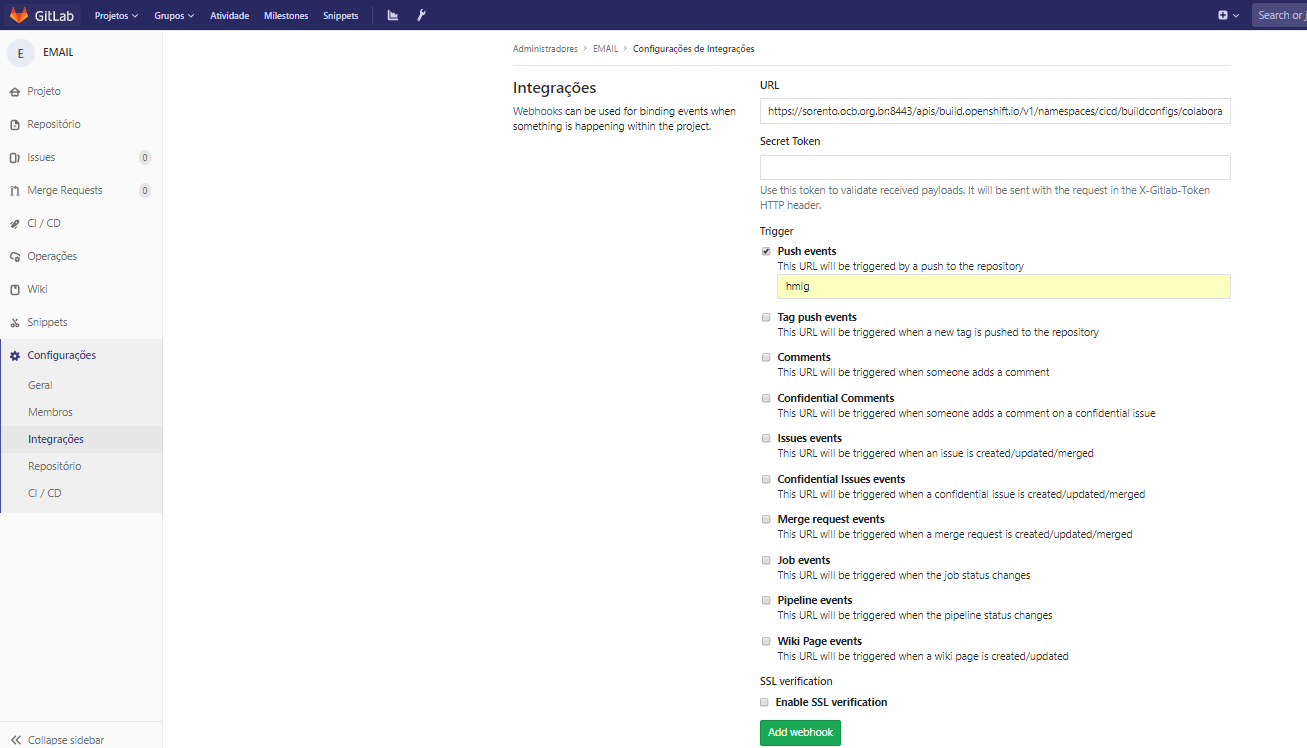
### Configurando a integração do Openshift com o GIT

Acessar o projeto no **GITLAB** e, no menu, Configurações/Integrações, colar a “Webhook URL” do Openshift, no campo “URL”.

O valor do campo “Push Events” deve ser “hmlg”.

Desabilitar o campo “Enable SSL verification”.

E acionar o botão “Add webhook” para salvar a configuração de integração.



## Iniciando o Pipeline

Para iniciar o Pipeline, bastará dar um *push* do código-fonte para a *branch* “hmlg” ou, no Openshift, utilizar o botão “Start Pipeline”.

