

Documentación Pipelines

Cliente: Banco Bolivariano de Ecuador

Proyecto: Integración continua para las líneas
24Móvil y OnBoarding

Equipo DevOps

Aprovisionamiento Pipelines Proyectos

3

1.1. Estructura de Directorios

3

1.2. Flujos de trabajo

3

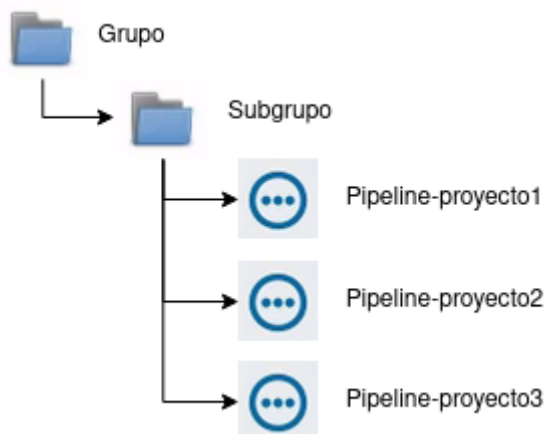
1.3. Ejecución Pipeline

7

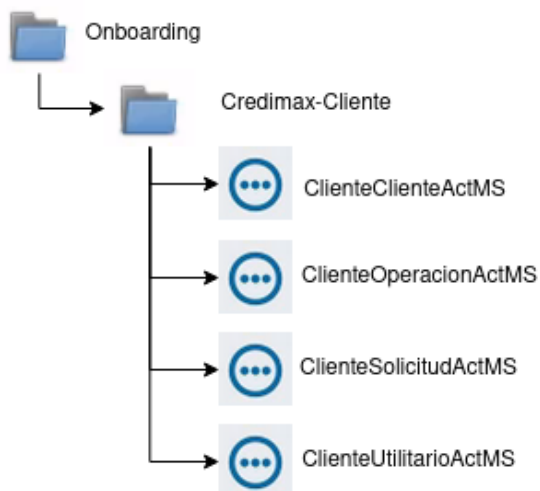
1. Aprovisionamiento Pipelines Proyectos

1.1. Estructura de Directorios

En la plataforma de Jenkins se realizó el aprovisionamiento de los pipelines correspondientes a los proyectos de los grupos de trabajo Onboarding y 24MovilRetail, quedando la estructura de directorios de la siguiente manera:



Ejemplo: ubicación de los pipelines de los proyectos del subgrupo **Credimax-Cliente**.



1.2. Flujos de trabajo

Para los proyectos Onboarding y 24MovilRetail se definieron los siguientes flujos de trabajo:

1. Proyectos ubicados en los subgrupos con sufijo "Microservicios" y "Cliente"

Se definió el siguiente ciclo de integración continua para estos proyectos:

- **Descarga de código fuente:** Jenkins realiza la descarga del código fuente almacenado en el repositorio del proyecto en Gitlab.

- **Compilación del código:** Utilizando maven versión 3.8.4, se ejecutará el siguiente comando para compilar el proyecto:

```
mvn package -Dmaven.test.skip=true -Dmaven.wagon.http.ssl.insecure=true
-Dmaven.wagon.http.ssl.allowall=true
-Dmaven.wagon.http.ssl.ignore.validity.dates=true
```

- **Ejecución de pruebas unitarias:** Utilizando maven versión 3.8.4, se ejecutará el siguiente comando para ejecutar las pruebas unitarias del proyecto:

```
mvn test
```

- **Ejecución de análisis estático de código:** El código es analizado con base a las reglas definidas en la plataforma de Sonarqube para cada lenguaje. Dependiendo del resultado del análisis, Sonarqube retorna una respuesta a Jenkins para continuar o detener el proceso.

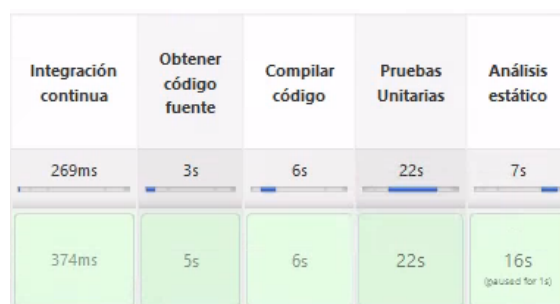
El resultado del análisis y el reporte de los hallazgos encontrados queda almacenado en la plataforma de Sonarqube.

Nota: Cada proyecto que se ejecuta desde la plataforma de Jenkins crea su propio proyecto, no es necesario crear el proyecto manualmente en la plataforma de Sonarqube.

Además, se construye el archivo sonar-project.properties con los siguientes datos:

```
sonar.projectName = <nombre proyecto>
sonar.projectKey = <nombre proyecto>
sonar.sources = src/main
sonar.projectBaseDir = ./
sonar.test = src/test
sonar.java.binaries = target/classes
```

Flujo de integración continua desde la plataforma de Jenkins



Flujo de despliegue continuo (pendiente por definir)

Declarative: Tool Install	Integración continua	Obtener código fuente	Compilar código	Pruebas Unitarias	Análisis estático	Ciclo de Producción	Aprobar Despliegue a Producción	Despliegue a Producción	Declarative: Post Actions
210ms	269ms	3s	6s	22s	7s	45ms	70ms	78ms	330ms
268ms	374ms	5s	6s	22s	16s (paused for 1s)				

2. Proyectos ubicados en los subgrupos "BaseDeDatos" y subgrupos con sufijo "Frontend"

Se definió el siguiente ciclo de integración continua para estos proyectos:

- **Descarga de código fuente:** Jenkins realiza la descarga del código fuente almacenado en el repositorio del proyecto en Gitlab.
- **Ejecución de análisis estático de código:** El código es analizado con base a las reglas definidas en la plataforma de Sonarqube para cada lenguaje; para los proyectos de base de datos se emplea el plugin para sql instalado en Sonarqube. Dependiendo del resultado del análisis, Sonarqube retorna una respuesta a Jenkins para continuar o detener el proceso.

El resultado del análisis y el reporte de los hallazgos encontrados queda almacenado en la plataforma de Sonarqube.

Nota: Cada proyecto que se ejecuta desde la plataforma de Jenkins crea su propio proyecto, no es necesario crear el proyecto manualmente en la plataforma de Sonarqube.

Para los proyectos se construye el archivo sonar-project.properties con los siguientes datos:

```
sonar.projectName = <nombre proyecto>
sonar.projectKey = <nombre proyecto>
sonar.projectBaseDir = ./
```

Flujo de integración continua desde la plataforma de Jenkins

Integración continua	Obtener código fuente	Análisis estático
72ms	2s	11s
76ms	1s	17s (paused for 2s)

Flujo de despliegue continuo (pendiente por definir)

Integración continua	Obtener código fuente	Análisis estático	Ciclo de Producción	Aprobar Despliegue a Producción	Despliegue a Producción	Declarative: Post Actions
72ms	2s	11s	46ms	69ms	69ms	320ms
76ms	1s	17s (paused for 2s)				

3. Proyectos ubicados en los subgrupos con sufijo "Config"

Se definió el siguiente ciclo de integración continua para estos proyectos:

- **Descarga de código fuente:** Jenkins realiza la descarga del código fuente almacenado en el repositorio del proyecto en Gitlab.

Flujo de integración continua desde la plataforma de Jenkins

Integración continua	Obtener código fuente
102ms	5s
102ms	5s

Flujo de despliegue continuo (pendiente por definir)

Integración continua	Obtener código fuente	Ciclo de Producción	Aprobar Despliegue a Producción	Despliegue a Producción
102ms	5s	0ms	0ms	0ms
102ms	5s			

Para todos los proyectos aún no se ha definido el flujo de despliegue continuo a ejecutar para el despliegue de los proyectos, por el momento los pipelines enviarán una notificación vía correo electrónico a quién corresponda realizar la aprobación para continuar con el proceso de despliegue.

Ejemplo notificación de solicitud de aprobación:

Solicitud de Aprobación Despliegue Proyecto: onboarding/Credimax-Microservicios /pruebalibreria
Externo
Recibidos x


info@bolivariano.com
para mí

mar, 21 dic, 19:03 (hace 6 días)
☆
↶
⋮

Se solicita la aprobación para realizar el despliegue al ambiente de producción del microservicio onboarding/Credimax-Microservicios/pruebalibreria.

Por favor ingrese a la siguiente URL para aprobar el despliegue:
<https://jenkinsdes.bolivariano.fin.ec/job/onboarding/job/Credimax-Microservicios/job/pruebalibreria/14/>

NOTA: Las tildes de este mensaje han sido omitidas expresamente para evitar cualquier tipo de alteracion en los caracteres del texto.

Nota: Si el pipeline presenta alguna falla en cualquier parte del proceso, se enviará una notificación vía correo electrónico a las personas correspondientes en cada proyecto, con información del pipeline donde se presentó el error.

Ejemplo de la notificación de fallo del pipeline:

Fallo Ejecución - Proyecto: SolicitudActMS
Externo
Recibidos x


info@bolivariano.com
para mí

9:54 (hace 6 horas)
☆
↶
⋮

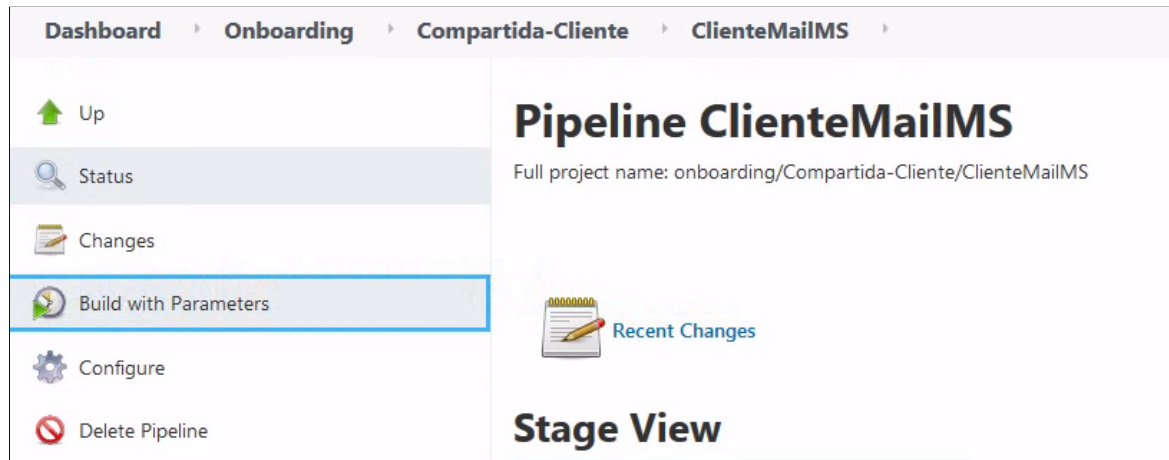
En la ejecución No.14 del proyecto SolicitudActMS se presentó un fallo.

Por favor ingrese a la siguiente URL para mayor información:
<https://jenkinsdes.bolivariano.fin.ec/job/onboarding/job/Credimax-Microservicios/job/SolicitudActMS/14/>

NOTA: Las tildes de este mensaje han sido omitidas expresamente para evitar cualquier tipo de alteracion en los caracteres del texto.

1.3. Ejecución Pipeline

Para ejecutar un pipeline seleccione la opción **Build with Parameters**, ubicada al lado izquierdo.



Los pipelines de los proyectos son parametrizados, es decir, que cuentan con un parámetro que se emplea para la ejecución del pipeline. El parámetro en este caso es **DESPLIEGUE_PRODUCCION**, dependiendo del valor de este parámetro se ejecutará solo el ciclo de integración continua o todo el proceso de despliegue continuo.

Pipeline ClienteMailMS

This build requires parameters:

☐ DESPLIEGUE_PRODUCCION

¿Desea desplegar hasta el ambiente de producción?

Build

Por defecto este parámetro se encuentra desactivado, por lo que el pipeline solo ejecutará el ciclo de integración definido para cada proyecto al dar clic en **Build**. Si se activa, se ejecutará el ciclo de producción (pendiente por definir), donde se solicita aprobación para continuar el proceso.

Pipeline ClienteMailMS

This build requires parameters:

☒ DESPLIEGUE_PRODUCCION

¿Desea desplegar hasta el ambiente de producción?

Build

Despliegue Automático

El proceso se gatillará a Gitlab, para que al momento de realizar un push sobre el repositorio del proyecto en la rama Desarrollo, el pipeline asociado al proyecto se ejecute automáticamente.