**PRUEBAS DE RENDIMIENTO EN PIPELINE MICROSERVICIOS**

**INFORMACIÓN GENERAL:**

Para la implementación de las pruebas de Rendimiento en el pipeline de microservicios fue necesario crear diferentes variables que serán consideradas dentro del proceso de ejecución y mediante las cuales se determinarán los estados de cada ejecución del pipeline.

Las variables creadas para ser utilizadas en el pipeline (IC y Release) junto con XL-Release en esta implementación son las siguientes:

* **vu** = Representa el valor del virtual user, mediante el cual se configura la cantidad de threads que utilizará jmeter para realizar la prueba de rendimiento del microservicio.
* **ramUp** = Representa el valor del lapso en segundos que debe esperar jmeter para lanzar la ejecución de un nuevo thread. Por ejemplo: Si el thread tiene un valor de 10 y el rampup tiene un valor de 1, esto significa que por cada segundo Jmeter lanzará un Thread.
* **averageTimeLimit** = Representa el valor del límite promedio de tiempo de respuesta en segundos que no debe ser superado por la ejecución de las pruebas de rendimiento sobre un microservicio. Se estipula que podría ser de 8 segundos, dependiendo de la complejidad del microservicio.
* **porcError** = Representa el porcentaje de error admitido que puede resultar del total de la ejecución de la prueba de rendimiento sobre un microservicio. Se estipula que podría ser del 10%, pero como en el caso anterior dependerá de la complejidad del microservicio y sus pruebas.

**Importante:**

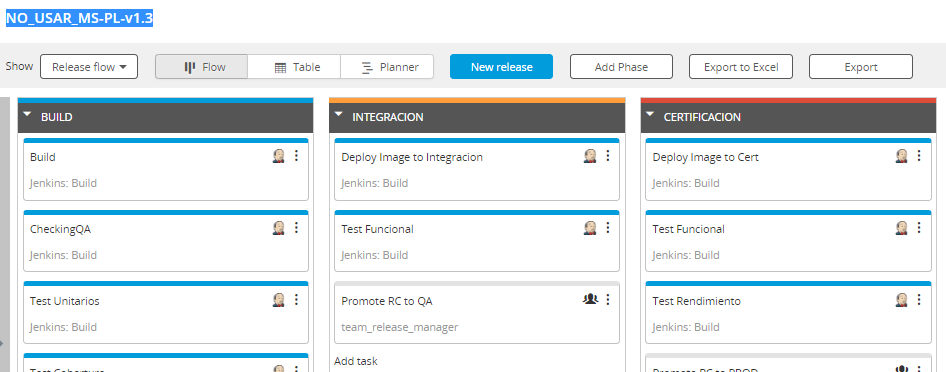
Para que el pipeline de microservicios pueda ejecutar correctamente las pruebas de Rendimiento en Jenkins Release, el archivo de configuración de las pruebas de Jmeter debe cumplir con las siguientes condiciones:

* Debe ser almacenado dentro de la carpeta “jmeter” en la raíz del repositorio del proyecto.
* Si dicho archivo tiene dependencia de otro archivo, por ejemplo un csv, éste debe ser almacenado en la misma carpeta “jmeter”.
* Debe poseer el nombre del proyecto microservicio que lo ejecuta. Por ejemplo: en el caso del microservicio [git@bitbucket.org:ejemploPruebas/**ms-pruebaRendimiento-neg**.git](mailto:git@bitbucket.org:ejemploPruebas/ms-pruebaRendimiento-neg.git), el nombre del archivo debe ser *“ms-pruebaRendimiento-neg.jmx”* para que sea reconocido para las pruebas de Rendimiento.

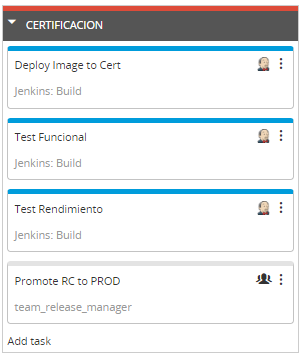
Si estas condiciones no se cumplen, el pipeline reportará la usencia del archivo jmx como advertencia y continuará con la ejecución del Job y los Stages restantes, si es el caso.

**EN XL-RELEASE:**

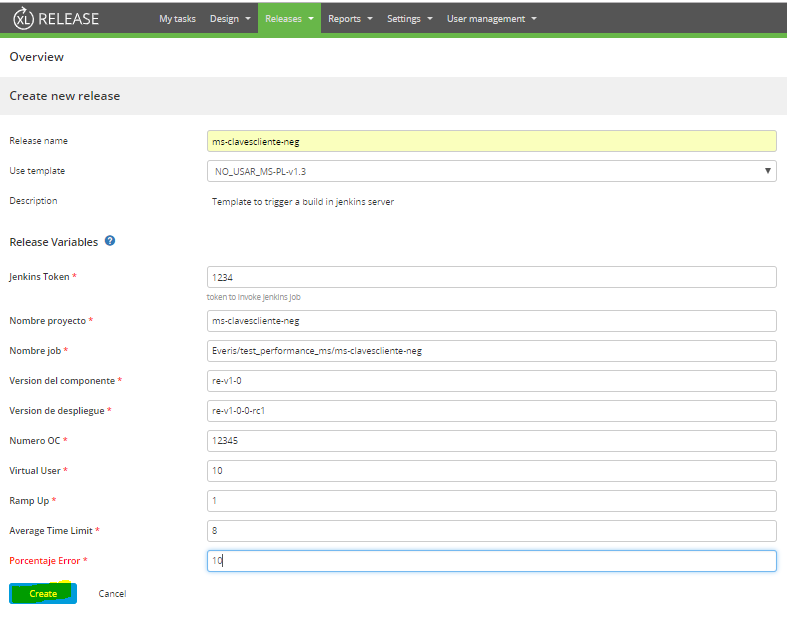
1. Para habilitar la implementación de las Pruebas de Rendimiento en XL-Release fue necesario crear un nuevo template, que actualmente se llama “NO\_USAR\_MS-PL-v1.3”, pero que cuando sea liberada esta implementación pasará a llamarse “MS-PL-v1.3”, reemplazando así a la actual versión del template de microservicios.



1. El nuevo template de microservicios contiene en la fase *“CERTIFICACION”* un nuevo task llamado ***“Test Rendimiento”***, el cual fue parametrizado con las variables existentes del template más las variables explicadas anteriormente en este documento.

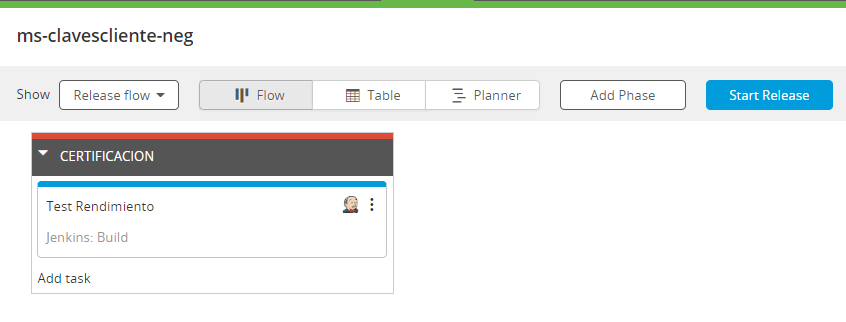


1. Al comenzar un nuevo release presionando “New Release”, se deberá completar los campos existentes anteriormente más los nuevos campos relacionados con las variables explicadas anteriormente.



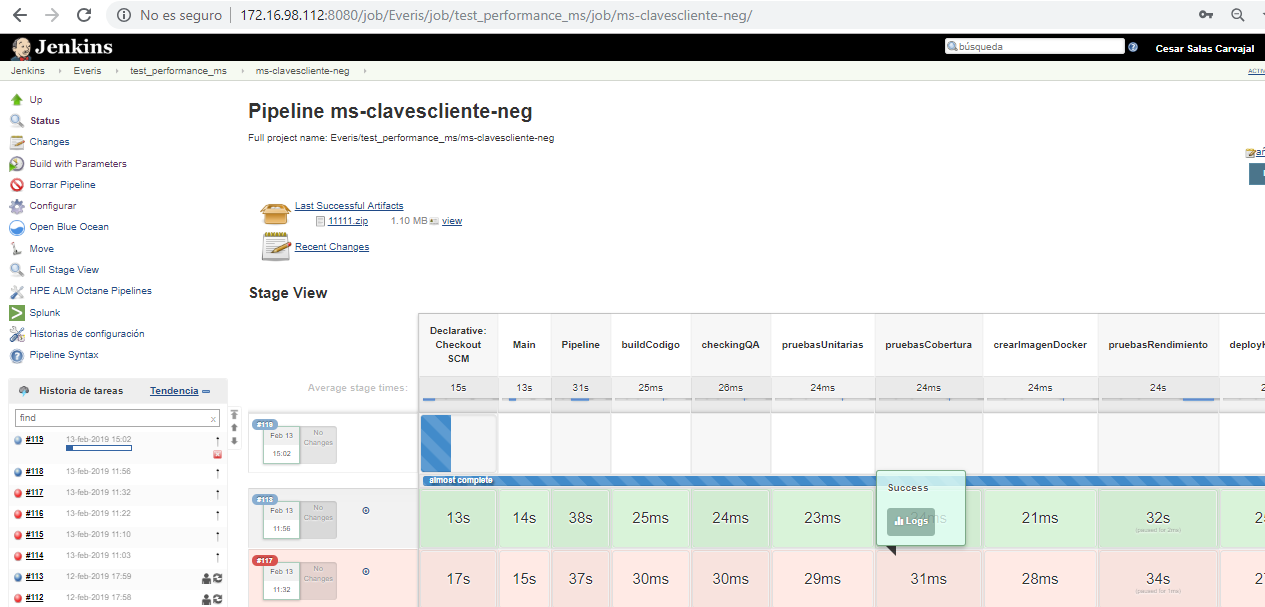
Los datos de prueba utilizados en esta ejecución corresponden a:

1. Release name = ms-clavescliente-neg
2. Nombre proyecto = ms-clavescliente-neg
3. Nombre Job = Everis/test\_performance\_ms/ms-clavescliente-neg
4. Version del componente = re-v1-0
5. Version de despliegue = re-v1-0-0-rc1
6. Numero OC = 12345
7. Virtual User = 10
8. Ramp Up = 1
9. Average Time limit = 8
10. Porcentaje Error = 10
11. Con lo anterior ingresado, se procede a ejecutar la tarea de XL-Release y esperar a que el Job de Jenkins 112 comience.

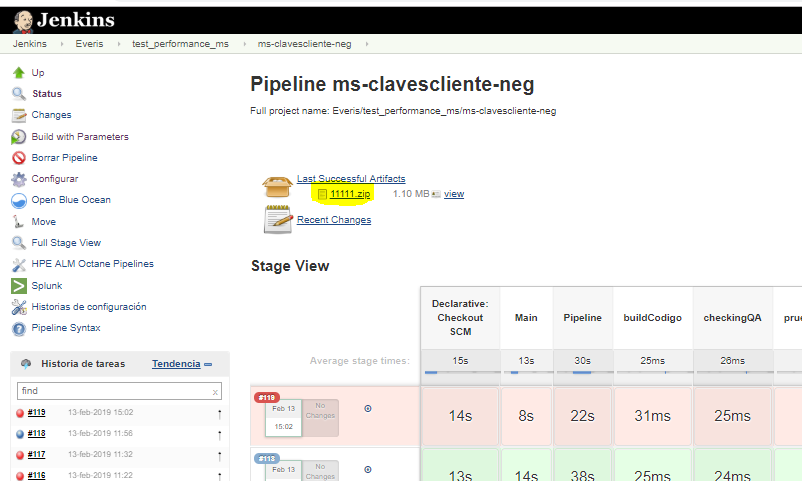


**EN RELEASE:**

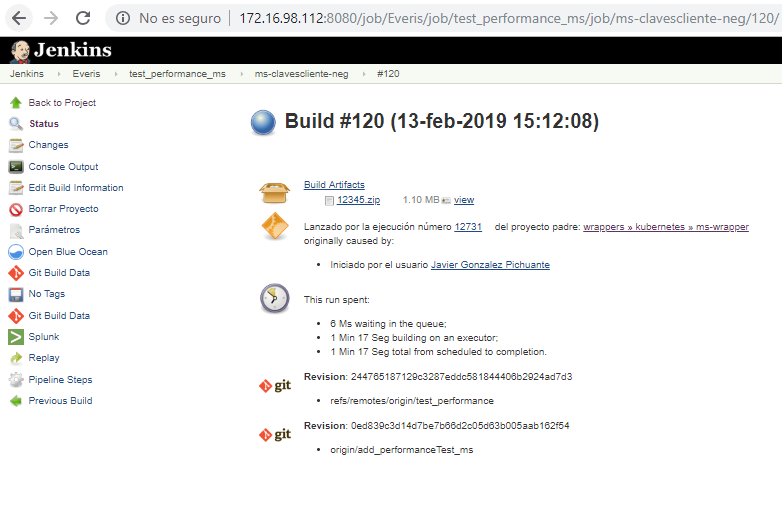
1. En Jenkins Release (Jenkins 112) comenzará el Job asociado a la ejecución realizada anteriormente en XL-Release, recibiendo los parámetros para realizar las pruebas de rendimiento.



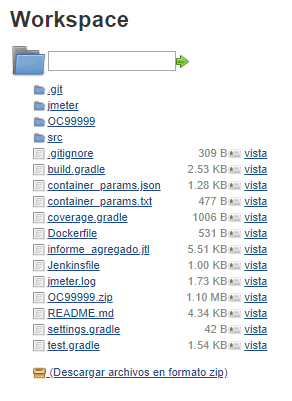
1. En caso que las pruebas de rendimiento del microservicio fallen, podría deberse a que se sobrepasó el límite de tiempo promedio total de respuesta (averageTimeLimit) o el porcentaje de error aceptado (porcError). Lo anterior, no sería impedimento para que el pipeline disponibilice el reporte HTML en formato OCNumero.zip en el estado del Job, como se muestra, de tal manera que sea descargado y analizado por la célula o personal que lo requiera.



1. Si las pruebas de rendimiento del microservicio son exitosas también podrá visualizarse el reporte HTML en formato OCNumero.zip en el estado del Job.



1. En Ambos casos, para el análisis por parte de los interesados, el pipeline genera archivos relacionados con la ejecución de las pruebas de rendimiento a través de jmeter. Dichos archivos se encuentran en el workspace:



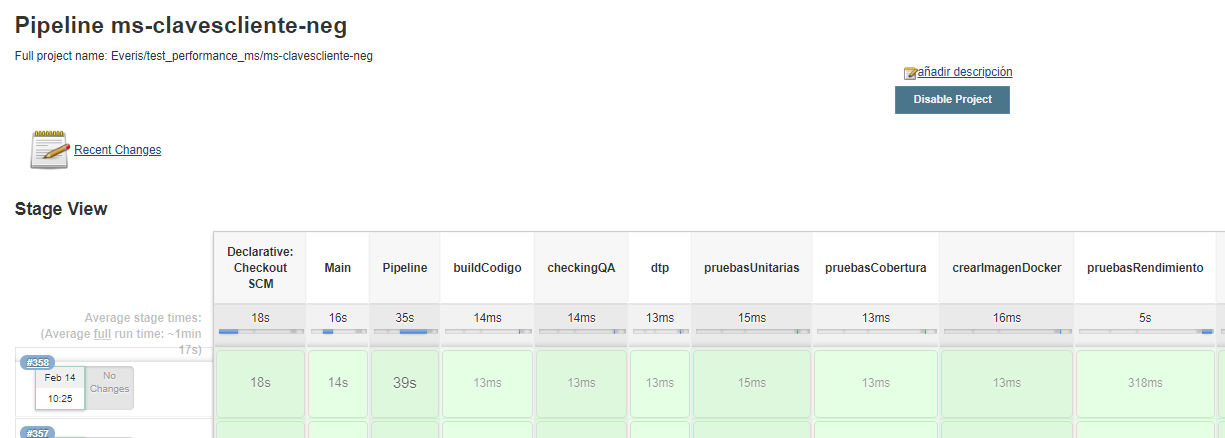
* Folder “OC99999”: carpeta creada con el número de OC ingresado en XL-Release. Contiene el reporte grafico HTML de las pruebas de rendimiento realizadas.
* File “OC99999.zip”: archivo zip creado por el pipeline para ser expuesto en la ejecución del Job y descargado desde jenkins. Contiene la misma información que se muestra en el folder “OC99999”.
* Folder “jmeter”: carpeta donde la célula debe dejar su archivo jmx y asociados a él como el archivo “testdata.csv”. Además, el pipeline de microservicios utiliza esta carpeta para almacenar los archivos de reportes del jmeter que va generando en su proceso.
  + File “nombre\_proyecto\_modified.jmx”: es el archivo resultante de la modificación que se realiza al jmx original “nombre\_proyecto.jmx” con los parámetros “vu” y rampup” ingresados en la ejecución de XL-Release.
  + File “test\_results.jtl”: contiene los resultados de las pruebas realizadas por jmeter, base de otros reportes y el informe gráfico contenido en el archivo zip.
  + File “aggregateResults.csv”: contiene datos más estadísticos utilizados para la toma de decisiones.

**EN IC**

1. Para ejecutar las pruebas de rendimiento en Jenkins IC (Jenkins 110), se ha considerado de forma automática e interna el seteo de las variables “vu” y “rampup” con valor igual a 1. Además, las variables “averageTimeLimit” y “porcError” no son consideradas en la ejecución de las pruebas en IC. La configuración anterior se justifica debido a que en IC las pruebas de rendimiento se ejecutarían solo para validar el archivo jmx con sus pruebas y su correcta lectura, por lo que no representaría una prueba de carga o rendimiento real.
2. Por lo anterior, cuando se ejecuta el Job de un microservicio, para realizar solo las pruebas de rendimiento se deberá ingresar el nombre del stage “pruebasRendimiento” en el parámetro solicitado y presionar el botón “ejecución”.



1. Jenkins procederá a la ejecución del pipeline de microservicios y si el estado del archivo jmx es correcto la prueba se ejecutará correctamente.



1. Al igual que en Jenkins Release, en este caso también se disponibilizará los archivos generados por jmeter en la carpeta “jmeter” del repositorio dentro del workspace, el cual contendrá los mismos archivos descritos en la sección del Jenkins Release.

