

Prueba Consultor DevOps Credibanco

Se realiza el aplicativo web utilizando Angular 16, la aplicación muestra el mensaje de bienvenida, la fecha y hora actual.

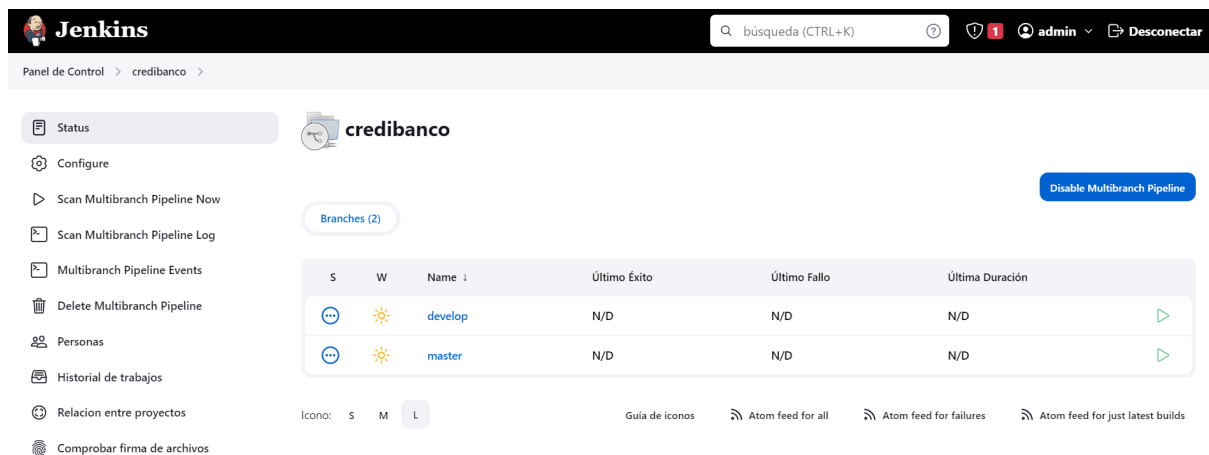
<https://github.com/rovzu/credibanco>

El repositorio tiene la rama Develop y Master.

La rama master tiene restricciones de no escritura y solo acceso vía Pull Request.

La rama develop está abierta para los nuevos desarrollos.

Se configuró en Jenkins un pipeline tipo Multibranch que a través de los webhooks de github ejecuta el pipeline cuando se tiene algún cambio sobre la rama.

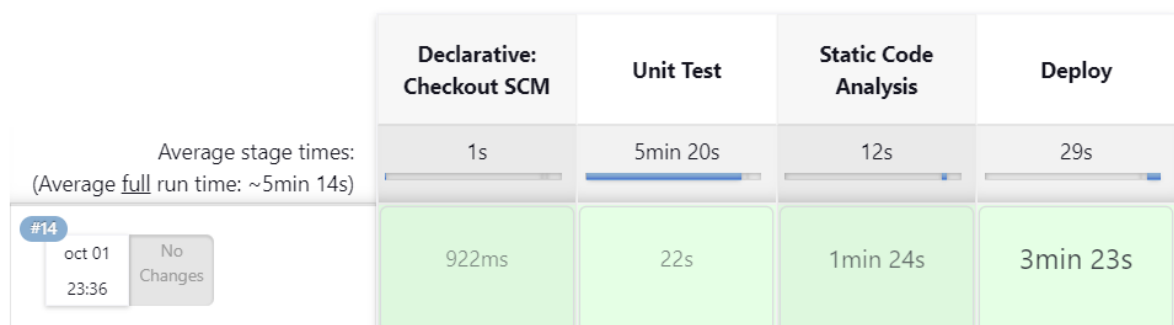


S	W	Name	Último Éxito	Último Fallo	Última Duración
...	☀	develop	N/D	N/D	N/D
...	☀	master	N/D	N/D	N/D

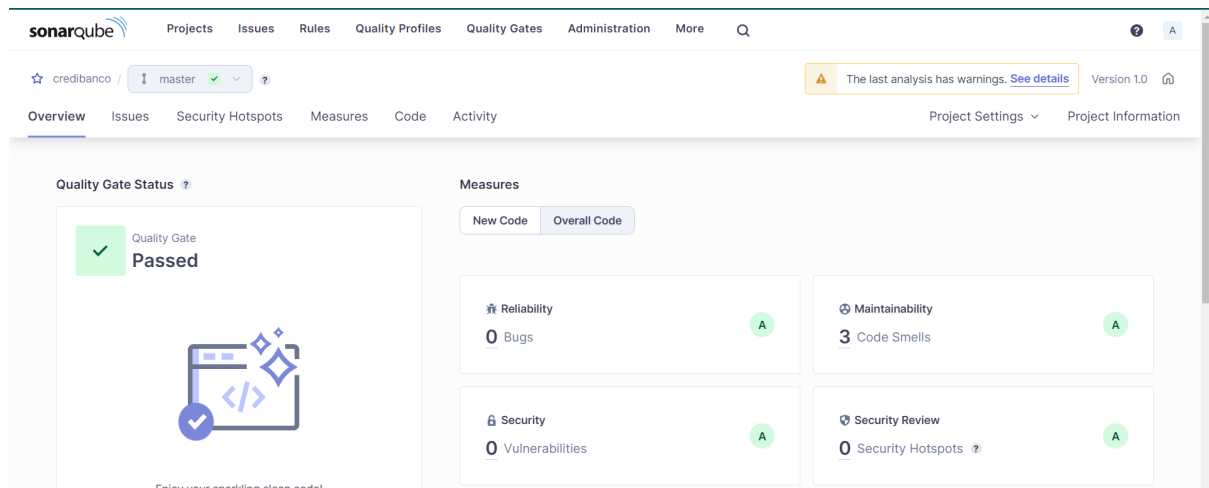
La rama develop tiene el pipeline asociado de integración continua con los siguientes stages: Pruebas unitarias, Análisis de código estático con Sonar, Compilación de imagen docker de la aplicación y publicación de imagen en el registry de docker hub.

La imagen se está publicando en el siguiente registry de DockerHub <https://hub.docker.com/repository/docker/santiagoapps4/credibanco/>

Stage View



Se tiene SonarQube como herramienta on Premise de análisis de código estático.



La rama de Máster tiene el pipeline asociado de Despliegue continuo actualizando el ambiente de AWS Elastic BeanStalk con la nueva imagen desplegada.

Infraestructura Asociada.

El despliegue se hizo sobre el servicio PaaS de AWS Elastic BeanStalk utilizando una VPC con máscara de red /29 y solo permitiendo acceso http a través del puerto 80.

No se cuenta con escalamiento para evitar costos adicionales en la capa Freetier. Se habilita el monitoreo de infraestructura y de aplicación a través del Servicio de CloudWatch.

Pruebas de performance

Se realiza un cálculo de pruebas de performance suponiendo que se tendrán 10 peticiones por segundo, ejecuta una prueba base donde se evidencian tiempos de respuesta mínimos de 81 milisegundos y máximo de 249 milisegundos.

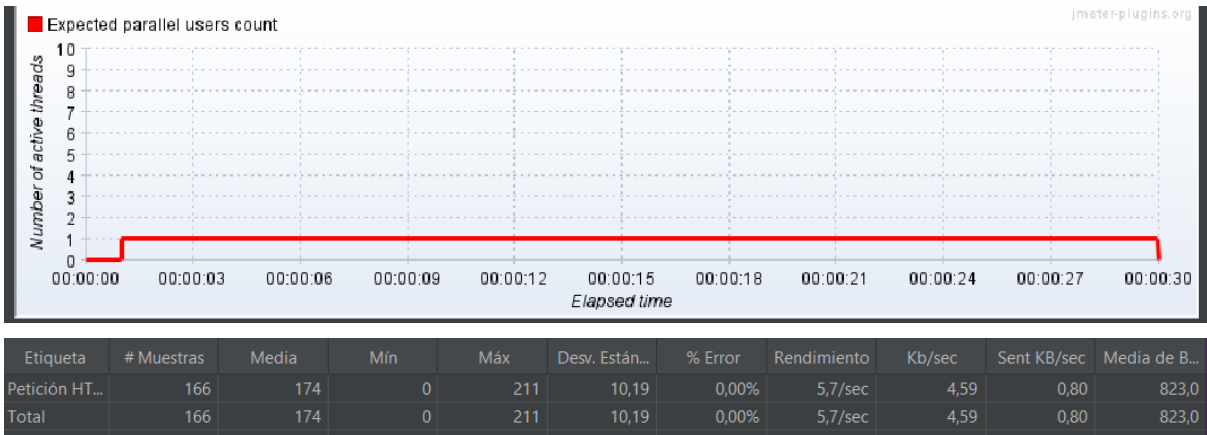
Se realiza el cálculo de concurrencia usando la fórmula de probabilidad Poisson para calcular cuál es la cantidad de usuarios probables que se tendrán en un periodo de tiempo.

La carga, es decir, el 100% de la carga esperada se debe ejecutar con 8 hilos.

La línea base, es decir el 10% de la carga esperada se debe ejecutar con 1 hilo.

La prueba de estrés 125%, se debe ejecutar con 10 hilos.

Para la prueba de línea base se observa el siguiente comportamiento.



Para la prueba de carga se observa el siguiente comportamiento.

