

PROFESOR: Bastian Ruiz Garay

FECHA ENUNCIADO: 12 de agosto de 2024

INTRODUCCIÓN

Hoy en día, las empresas que desarrollan software necesitan encontrar mecanismos que les permitan integrar código de forma constante, construir la aplicación sin contratiempos y desplegarla en los ambientes respectivos. Todo esto debe realizarse de forma automática y recibiendo retroalimentación en el caso que algo dentro de este proceso falle. El objetivo de este taller es el desarrollo de un pipeline a escala pequeña que automatice este proceso.

DESCRIPCIÓN DEL TALLER

Para este trabajo, se le proveerá una aplicación web que deberá desplegar en un servidor utilizando un pipeline CI/CD. La aplicación se adjuntará en un archivo comprimido (.zip) junto a este enunciado.

La aplicación web consiste en el siguiente microservicio:

- **Microservicio de Gestión de Usuarios:** Construido con NestJS. Posee funcionalidades de Login y funciones CRUD del usuario. También posee pruebas unitarias.

El trabajo consiste en recrear una arquitectura similar a la utilizada en un ambiente DevOps real. Para esto, deberán establecer dos servidores:

- **Servidor CI/CD:** En este servidor deberá establecer los servicios que ejecutarán el pipeline con las tareas automáticas de testing, building y deploy.
- **Servidor de Despliegue:** Aquí deberá desplegar el microservicio que pasará por el pipeline.

En el servidor de CI/CD, deberán instalar un servicio de ejecución de tareas automáticas (como Jenkins) para desarrollar un pipeline que tome el código del microservicio y verifique su correcto funcionamiento. Posteriormente, deberá usar un mecanismo para generar un ejecutable y enviarlo al servidor de despliegue (como Docker con DockerHub). Este pipeline debe ejecutarse automáticamente cuando se realice un cambio en el código, por lo que deberá crear un repositorio en GitHub, subir el código fuente base, y subir un cambio en una rama independiente para ejecutar la acción.

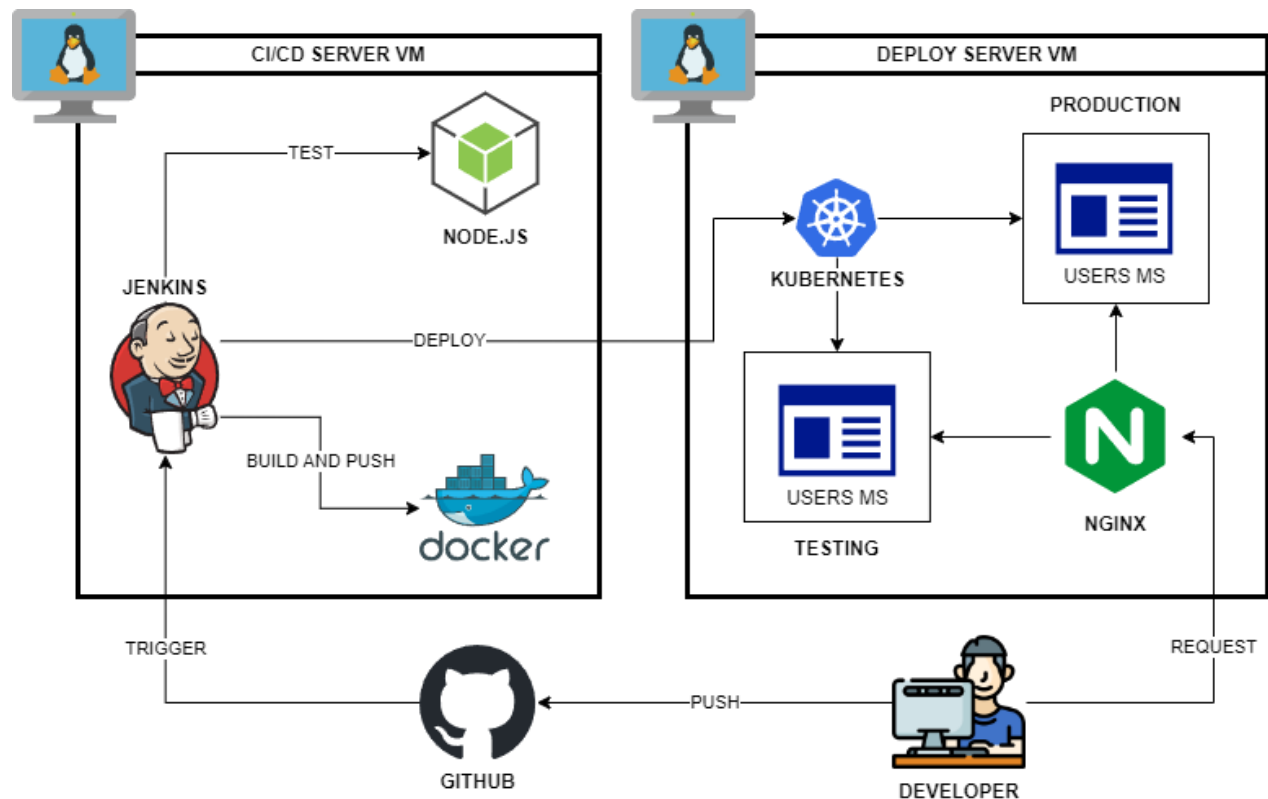
En el servidor de despliegue, deberá definir dos ambientes separados: uno para despliegue de aplicaciones listas para producción (utilizando la rama *main* de los repositorios) y otro ambiente para realizar testing (para las ramas que se mezclarán con *main*, usando alguna tipología compatible con trunk-based development). Puede utilizar servicios como Nginx para realizar el enrutamiento de puertos.

Los servidores puede implementarlos de la forma que desee. Se sugiere una de las siguientes infraestructuras:

- Utilizar Servicios en la Nube (como AWS EC2, Azure VM o Google Cloud Compute Engine).
- Utilizar 2 máquinas virtuales dentro del mismo computador (o dos computadores conectados en red).
- Utilizar Contenedores de Docker (para el servidor CI) y Minikube (separando los ambientes por namespace).

Deberá diagramar y explicar el modelo de conectividad de sus servicios, indicando los puertos de comunicación y las IPs de cada máquina, tomando en cuenta desde el momento en que se realiza una petición PUSH al servidor de GitHub, hasta el momento en que un usuario puede hacer una solicitud a los servicios desplegados.

En la siguiente imagen se muestra un ejemplo de la arquitectura a seguir.



En esta arquitectura, un desarrollador hace una solicitud PUSH a GitHub con los cambios realizados al repositorio. Este cambio genera un trigger hacia el servicio de Jenkins para la ejecución del pipeline del microservicio y ambiente respectivo. En primer lugar, Jenkins ejecuta las pruebas del microservicio, y luego construye la imagen de Docker y la envía a DockerHub.

Posteriormente, Jenkins envía la orden al servidor de despliegue al servicio de Kubernetes (k8s) para obtener la imagen producida y desplegarla en el ambiente correspondiente.

Por último, los usuarios y los desarrolladores pueden realizar una solicitud al servidor de despliegue para solicitar un servicio. Esta solicitud, según el puerto o subdominio, es interpretado por el servicio de Nginx y redirige la solicitud al ambiente y microservicio correspondiente.

Cabe destacar que el servicio desplegado (tanto en producción como en testing) deben exponer una URL (mediante IP o nombre de dominio) para ser consultados desde cualquier navegador instalado en la máquina host, o cualquier dispositivo en el caso de despliegue en la nube o apertura de puertos desde el dispositivo que aloja la máquina virtual.

EVALUACIÓN DEL TALLER

Este trabajo deberá ser entregado mediante un informe, donde se explique cada uno de los pasos realizados para generar la infraestructura y los pasos para probar su funcionamiento. Adicionalmente, deberá elaborar un video (de no más de 20 minutos) donde muestre el paso a paso de la implementación y las pruebas realizadas. Se pueden realizar cortes o aceleramiento del video en las partes de espera, como las instalaciones y los despliegues. Este video servirá como prueba y respaldo de lo desarrollado en el informe.

Ambos entregables deberán enviarse por correo electrónico como máximo el día **26 de agosto de 2024 hasta las 23:59**. Deberá entregar el informe en formato PDF, y el enlace del video donde se encuentre alojado (como hipervínculo dentro del informe o dentro de la entrega en un archivo de texto o similar). No se aceptarán entregas después de la fecha y hora indicada. **Si el video no se puede reproducir, el informe viene en otro formato que no sea PDF u ambos, la nota máxima a la que podrán optar como taller será un 4,0.**

El informe será evaluado utilizando la siguiente tabla:

ITEM	PUNTAJE	DESCRIPCIÓN
Estructura	3	El informe cumple con el estándar de estructura según formato UCN (Portada con logos e información del curso y grupo, Índice de contenidos, Índice de figuras, Introducción, Contenido, Conclusión, Bibliografía en APA y Anexos si los tiene).
Formato	3	El informe sigue un formato uniforme en base a las normativas indicadas por la escuela (Véase formato Informe de Práctica).
Ortografía y gramática	2	El informe no presenta faltas ortográficas, problemas de gramática ni textos sin sentido.
Introducción	2	El informe presenta una introducción coherente con los contenidos de la asignatura y el desarrollo del trabajo.
Recursos utilizados	2	Dentro del informe se indican los recursos tecnológicos de software (aplicaciones o servicios en la nube) con su descripción y uso en el trabajo.
Diseño arquitectura DevOps	4	En el informe se describe la arquitectura que se utilizará para la implementación del sistema completo, indicando los servidores, redes, puertos, herramientas y plataformas. Se presenta un gráfico o diagrama con la arquitectura.
Implementación en servidor CI/CD	6	Se mencionan los pasos para la implementación del servidor CI/CD, indicando los recursos, conexiones, limitaciones y justificación de las decisiones tomadas. Se describe la creación de los pipelines. Presentan imágenes mostrando el desarrollo del proceso.
Implementación en servidor de despliegue	6	Se mencionan los pasos para la implementación del servidor de despliegue, indicando los recursos, conexiones, limitaciones y justificación de las decisiones tomadas. Se describe la conexión con el servidor de CI/CD, los scripts utilizados, la definición de los ambientes de Producción y Testing y la forma de desplegar el microservicio. Presentan imágenes mostrando el desarrollo del proceso.
Ejecución de pruebas	4	Se muestra el proceso de prueba del servicio, desde el momento en que se realiza un push al repositorio, pasando por el servidor CI/CD, ejecutando las tareas de testing, building y deploy en el servidor de despliegue, además de mostrar el acceso al servicio mediante una ruta expuesta (IP o URL) en ambos ambientes.
Conclusión	2	El informe cuenta con una conclusión del trabajo realizado, describiendo el cumplimiento de objetivos y las enseñanzas obtenidas.

Por último, la nota final del taller será calculada utilizando la siguiente tabla:

EXIGENCIA	PUNTAJE TOTAL	PUNTAJE 4.0
60%	34	20,4

OTROS

- Si utilizan servicios en la nube para implementar su infraestructura, verifiquen los costos de implementación para evitar sobrecargos, como también el tiempo en que cada máquina se mantiene encendida. **Ni el docente ni la Escuela de Ingeniería pagarán los costos de un servicio utilizado por los estudiantes.**
- Pueden utilizar cualquier herramienta de Inteligencia Artificial (como ChatGPT, Gemini, Copilot, etc.) para construir el informe y/o implementar la infraestructura. En dicho caso, se recomienda revisar el informe varias veces para evitar problemas con los textos generados.
- Cualquier consulta respecto a la evaluación y al enunciado en general, pueden realizarla mediante correo electrónico (bastian.ruiz@ce.ucn.cl).
- Cualquier cambio respecto a este enunciado será notificado mediante correo electrónico.