

Índice

Introducción.....	Pag. 2
Gradle.....	Pag. 3 y 4
Git.....	Pag. 5 y 6
Nagios.....	Pag. 7 y 8
Jenkins.....	Pag. 9 y 10
Docker.....	Pag. 11 y 12
Conclusión.....	Pag. 13
Anexos.....	Pag. 14
Bibliografía.....	Pag. 15

Introducción

En este trabajo estaremos viendo cinco Framework que se pueden utilizar para trabajar con DevOps en el cual hay muchas opciones pero estos cinco fueron los que me parecieron mas interesantes y mejores, explicare algunos detalles sobre su origen, historia, las ventajas y desventajas que estos proporcionan aparte de que como instalarlos y la importancia que tienen en este mundo como ejemplo tomemos el primero que es Gradle es una herramienta que permite la automatización de compilación de código abierto, la cual se encuentra centrada en la flexibilidad y el rendimiento lo cual es perfecto para un trabajo optimo en las áreas de DevOps.

Gradle

Un Framework para trabajar con DevOps es Gradle, este se originó por varios desarrolladores. Este fue lanzado por primera vez en 2007, y en 2013, fue adoptado por Google como el sistema de compilación para proyectos de Android.

Este no ha tenido una historia muy larga, pero A partir de 2016, los complementos iniciales se centraron principalmente en Java, desarrollo e implementación de Groovy y Scala, pero hay más idiomas y flujos de trabajo de proyectos en la hoja de ruta.

Gradle se ejecuta en la JVM y debe tener instalado un Kit de desarrollo de Java (JDK) para usarlo. Esto es una ventaja adicional para los usuarios familiarizados con la plataforma Java, ya que puede usar las API Java estándar en su lógica de compilación, como los tipos de tareas y complementos personalizados. También facilita la ejecución de Gradle en diferentes plataformas. Varios IDE principales le permiten importar compilaciones de Gradle e interactuar con ellos: Android Studio, IntelliJ IDEA, Eclipse y NetBeans, este también tiene soporte para la construcción de archivos necesarios en Visual Studio.

La importancia de este se deriva en que Gradle es el siguiente paso evolutivo en las herramientas de compilación basadas en JVM. Se basa en las lecciones aprendidas de herramientas establecidas como Ant y Maven y lleva sus mejores ideas al siguiente nivel.

Las ventajas que este proporciona son:

- Gradle te da convenciones, pero aún te da el poder de anularlas fácilmente.
- Los archivos de compilación de Gradle son menos detallados ya que están escritos en groovy.
- Proporciona DSL muy agradable para escribir tareas de compilación.
- Tiene muchos complementos buenos y un ecosistema vibrante

Las Desventajas que este proporciona son:

- pobre integración con eclipse
- El caché de dependencia de Gradle no es portátil . Si lo copia entre máquinas, se romperá en la mayoría de los casos.
- La configuración del repositorio por dependencia no es posible

Las empresas que utilizan este Framework son Netflix, Lyft, Craftbase, Henge, etc.

Gradle tiene compatibilidad con muchas herramientas de terceros como Sonarqube, Buddy, Liquibase, Flyway entre otros.

Para la instalación de este primero Cree un nuevo directorio C:\Gradle con el Explorador de archivos.

Abra una segunda ventana del Explorador de archivos y vaya al directorio donde se descargó la distribución de Gradle. Haga doble clic en el archivo ZIP para exponer el contenido. Arrastre la carpeta de contenido gradle-6.5 a su C:\Gradle carpeta recién creada.

Alternativamente, puede desempaquetar el ZIP de distribución de Gradle C:\Gradle utilizando una herramienta de archivo de su elección.

Y para configurarlo En el Explorador de archivos, haga clic derecho en el icono This PC (o Computer), luego haga clic en Properties-> Advanced System Settings-> Environmental Variables.

En System Variables seleccionar Path, luego haga clic Edit. Agregar una entrada para C:\Gradle\gradle-6.5\bin. Haga clic en Aceptar para guardar.

Git

Otro Framework que se utiliza para el DevOps es Git, Git es un sistema de control de versiones distribuido que, como ya hemos introducido, fue creado por Linus Torvalds en 2005 para ser utilizado en el desarrollo del kernel de Linux.

Un poco de su historia de git. El kernel de Linux es un proyecto de software de código abierto con un alcance bastante amplio. Durante la mayor parte del mantenimiento del kernel de Linux (1991-2002), los cambios en el software se realizaban a través de parches y archivos. En el 2002, el proyecto del kernel de Linux empezó a usar un DVCS propietario llamado BitKeeper.

En el 2005, la relación entre la comunidad que desarrollaba el kernel de Linux y la compañía que desarrollaba BitKeeper se vino abajo y la herramienta dejó de ser ofrecida de manera gratuita. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron mientras usaban BitKeeper.

Tecnologías que se relacionan con Git Hay dos tipos principales y comunes de sistema de control de versiones:

El sistema de control de versiones centralizado (CVCS)

El sistema de control de versiones descentralizado o distribuido (DVCS)

El CVCS utiliza un servidor central y almacena todos los archivos. Funciona en un único repositorio y permite la colaboración en equipo. Es un servidor central al que todos los miembros del equipo tienen acceso.

Con DVCS, no es obligatorio confiar en un servidor central para almacenar versiones de los archivos del proyecto. Cada miembro del equipo tiene un clon del proyecto o archivo, el repositorio. Por lo tanto, cada uno puede efectuar algunas modificaciones y luego compartirlas con los demás en un servidor.

¿Por qué es importante utilizar Git?

Git es el sistema de control de versiones más utilizado. Git rastrea los cambios que realiza en los archivos, para que tenga un registro de lo que se ha hecho y puede volver a versiones específicas si alguna vez lo necesita. Git también facilita la colaboración, permitiendo que los cambios de varias personas se fusionen en una sola fuente

Algunas de sus ventajas son:

- Modelo distribuido: esto significa que su trabajo es suyo. Puede dejar que otros vean solo lo que es necesario. No todo tiene que ser público.
- Ramificar y fusionar es fácil: ramificar es un paseo por el parque. Se siente como una parte natural del flujo de trabajo. Son baratos (rápidos y consumen muy poco espacio) para que pueda ramificarse cuando lo desee.
- El flujo de trabajo es flexible: en comparación con VCS centralizado, git tiene las cualidades que le permiten elegir su propio flujo de trabajo.

Algunas de sus desventajas son:

- Curva de aprendizaje empinada: muchos comandos con muchas opciones, algunos comandos no son intuitivos y necesitan un nivel de comprensión de lo interno de git, los comandos y los argumentos son inconsistentes hasta cierto punto
- Los archivos binarios son un gran no: si su proyecto tiene archivos que no son de texto que se actualizan con frecuencia (imágenes para sitios web o documentos de MS Office), entonces git se hincha y se vuelve lento. (Creo que todavía funciona mejor que la mayoría de los sistemas)

Empresas que utilizan Git son Shopify, Reddit, trivago, Glovo entre otros.

Git es compatible con Github ya que este viene del mismo Git y Github sirve principalmente para la creación de código fuente de programas de ordenador.

Para instalar Git, Una vez descargada, dale click al archivo con el formato *Git-version.exe*. Al inicio preguntará por permisos para ejecutar el instalador, a lo cual debemos responder que sí.

Como muchos de los instaladores en Windows, debemos de aceptar las opciones por defecto y darle Next (siguiente) a todo hasta que nos salga el botón de instalar.

Al finalizar la instalación, nos dará la opción de abrir Git Bash para finalizar.

Nagios

Como bien Ethan Galstad y un grupo de desarrolladores originalmente escribieron Nagios como NetSaint. A partir de 2015 mantienen activamente los complementos oficiales y no oficiales . Nagios es un acrónimo recursivo

Se trata de un software que proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los responsables correspondientes mediante (entre otros medios) correo electrónico y mensajes

Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix.

Nagios está licenciado bajo la GNU General Public License Version 2 publicada por la Free Software Foundation.

Este recurso esta basado en la tecnología de Nagios Core.

Esta puede ser un Framework algo antiguo pero cumple muy bien con su trabajo que es monitorear, es importante ya que este escala bien para configuraciones simples y estáticas y tiene un lenguaje de configuración simple que puede ser de mucha utilidad.

Ventajas de usar Nagios:

- Nagios es un software de código abierto. Es gratis de usar y editar.
- Tiene una configuración abierta que es fácil de agregar scripts personalizados para ampliar los servicios disponibles.
- Hay muchos dispositivos que el sistema Nagios puede monitorear. El requisito es un protocolo SNMP en ese dispositivo.
- Alertar, notificar o comentar el estado del sistema. Tiene una variedad de herramientas de alerta.
- Tiene muchos complementos y complementos que se pueden descargar y desarrollar de forma gratuita.

Desventajas:

- Muchas características no están disponibles en una versión gratuita de Nagios. Las funciones como los asistentes o el tablero interactivo están disponibles en Nagios XI, que es muy costoso.
- Hay muchos archivos de configuración que son muy difíciles de configurar.
- El núcleo de Nagios tiene una interfaz confusa.
- Nagios no puede monitorear el rendimiento de la red (usos de ancho de banda o disponibles).
- Nagios no puede administrar la red, solo monitorear la red.

Algunas empresas que usan Nagios son:

Uber, Dropbox, Henge, Twitch, entre otras. Este es compatible con Unix Operating systems (OSes).

Ahora para instalar Nagios primero tenemos que:

- hacer doble clic en el archivo "Nagwin_1.2.0_Installer.exe". Luego de un momento, aparecerá una ventana con el título "Nagwin 1.2.0 Setup". Haz clic en "Next" (siguiente).
- Examina el acuerdo de licencia de Nagwin/Nagios y haz clic en el botón "I Agree" (acepto) para continuar el proceso de instalación.
- Haz clic en el botón "Browse" (examinar) para seleccionar una carpeta de destino para Nagios, o haz clic en "Next" (siguiente) para utilizar la ubicación predeterminada de "C:\ Program Files\ICW".
- Escribe el nombre de cuenta y contraseña que la utilidad de instalación genere, o escribe tu propio nombre de usuario y contraseña en los campos "Service Account" (cuenta de servicio), "Type Password" (escribir contraseña) y "Confirm Password" (confirmar contraseña).
- Haz clic en el botón "Install" (instalar) para instalar Nagwin utilizando la configuración seleccionada. Luego de un momento, la utilidad de instalación muestra el mensaje "Nagwin 1.2.0 is installed on this machine now" (Nagwin 1.2.0 está instalada ahora en esta computadora). ¡Haz clic en "OK" (aceptar) y "Close" (cerrar) y Listo!

Jenkins

Jenkins fue originalmente desarrollado con el nombre Hudson. El desarrollo de Hudson empezó en el verano de 2004 en Sun Microsystems. Su primera versión fue publicada en febrero de 2005.

El 1 de febrero de 2011, Oracle dijo que continuarían con el desarrollo de Hudson y consideraron a Jenkins un fork en lugar de un cambio de nombre. Jenkins y Hudson continúan como proyectos independientes y considerando al otro proyecto como fork. El 7 de julio de 2016 se hizo pública la primera versión 2.x6 con soporte LTS.

Jenkins es un servidor de automatización open source escrito en Java. Está en la tecnología del proyecto Hudson y es, dependiendo de la visión, un fork del proyecto o simplemente un cambio de nombre.

Este es importante por que sirve para crear y probar sus proyectos de software continuamente, lo que facilita a los desarrolladores la integración de los cambios en el proyecto y facilita a los usuarios obtener una nueva versión.

Ventajas de utilizar Jenkins:

- Software independiente
- Se puede usar para muchos sistemas (que no sean AWS)
- Muchas opciones de configuración (por ejemplo, complementos)
- Se puede usar para configurar repositorios de código localmente

Desventajas

- Puede ser difícil de configurar.
- No es el mejor informe de errores.
- No está basado en la nube.
- Puede ser difícil de entender.
- Alta curva de aprendizaje.
- Tiene que mantener la infraestructura usted mismo.

Algunas empresas que utilizan este Framework son la NASA, Netflix, Facebook, Yahoo entre otras.

Es compatible con herramientas SCM, incluidas AccuRev, CVS, Subversion, Git, Mercurial, Perforce, Clearcase y RTC, y puede ejecutar proyectos basados en Apache Ant y Apache Maven, así como scripts de shell arbitrarios y comandos por lotes de Windows.

Para instalar Jenkins solo hay que seguir estos pasos.

1. Descarga la herramienta de la página oficial. Observa que puedes descargar Jenkins como paquete nativo para tu sistema operativo o en formato WAR. Para nuestro ejemplo, te recomiendo la segunda opción.
2. Ubica el WAR descargado (*war*) en la carpeta donde quieras instalar Jenkins.
3. Por último, sólo queda levantar el servidor. Para ello sitúate en la carpeta donde guardaste el WAR y ejecuta desde línea de comandos la instrucción `java -jar jenkins.war start`
4. Otra opción es que utilices un servidor de aplicaciones que tengas instalado para ejecutar el WAR, por ejemplo, Tomcat 8.
5. Si el servidor se inició correctamente, podrás acceder a Jenkins en la URL `http://localhost:8080`.

Docker

Docker Inc. fue fundada por Solomon Hykes y Sebastien Pahl durante el grupo de incubadoras de inicio Y Combinator Summer 2010 y se lanzó en 2011. Hykes comenzó el proyecto Docker en Francia como un proyecto interno dentro de dotCloud , una plataforma como servicio empresa.

Docker debutó al público en Santa Clara en PyCon en 2013. Fue lanzado como código abierto en marzo de 2013. En ese momento, utilizaba LXC como su entorno de ejecución predeterminado. Un año después, con el lanzamiento de la versión 0.9, Docker reemplazó LXC con su propio componente, que estaba escrito en el lenguaje de programación Go

La tecnología Docker usa el kernel de Linux y las funciones de este, como Cgroups y namespaces, para segregar los procesos, de modo que puedan ejecutarse de manera independiente.

Usar Docker es importante ya que Docker es un proyecto de código libre que ha ido convirtiéndose en una de las herramientas más utilizadas de los profesionales en desarrollo web y de aplicaciones o los administradores de sistemas debido a que proporciona muchas ventajas y brinda una facilidad de trabajo en base al concepto de contenedores.

Poco a poco Docker está cambiando la forma de desarrollar una aplicación o administrar un sistema, ya que nos permite encapsular características específicas y necesarias para un proyecto. Docker también nos ayuda a no malgastar el tiempo en la configuración de un entorno, y las dependencias necesarias de un sistema, ya que nos permite desplegarlo fácilmente.

Desventajas de usar Docker:

- Se requiere mínimo la versión de Kernel 3.8.
- Algunas versiones de Docker dan error debido a que se encuentran en constante desarrollo.
- Solo soporta a sistemas operativos Linux de arquitectura de 64 bits.
- Para Windows aún se encuentra en fase de desarrollo.

Ventajas:

- Retorno de la inversión y ahorro de costes.
- Estandarización y productividad.
- Eficiencia de imágenes de contenedor.
- Compatibilidad y mantenimiento más fácil.
- Simplicidad y configuraciones más rápidas.

Empresas que utilizan este Framework son Ebay, Spotify, Netflix, entre otras.

Este puede ser compatible con Frameworks de .NET.

Los pasos para instalar este son:

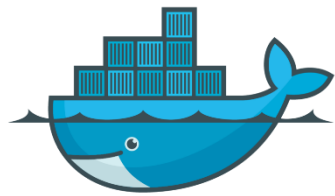
1. Primero para descargar el instalador de Docker para Windows tienes que logearte, solo precisas un nombre, un password y tu e-mail. Es gratuito.
2. Si ya tienes los pasos anteriores bien configurados es el momento de iniciar la instalación de Docker.
3. Ejecuta el archivo exe descargado y sigue los pasos normales de cualquier instalación en Windows.

Conclusión

En este trabajo pudimos crear un resumen de cada uno de los Framework elegidos por mí y abarcar lo mejor posible cada detalle que tiene como bien saber cada origen de donde vino ese Framework, quien lo creo, su historia y todo lo que paso para llegar a lo que es hoy, también vimos sus tecnologías, el por que de su importancia y el impacto que genera su utilización y cada ventaja y desventaja que tiene está.

¡Muchas Gracias!

Anexos



docker

Nagios[®]



git



Gradle



Jenkins

<https://mindmajix.com/devops-tools-and-frameworks#:~:text=Some%20of%20the%20widely%20used,Scrum%2C%20Sprint%2C%20and%20safe.&text=From%20the%20stage%20to%20development,automation%20of%20the%20delivery%20pipeline.>

Bibliografía

<https://openwebinars.net/blog/que-es-gradle/>

<https://es.wikipedia.org/wiki/Gradle>

<https://stackshare.io/gradle>

<https://codigofacilito.com/articulos/que-es-git>

<https://es.wikipedia.org/wiki/Git>

<https://stackshare.io/git>

<https://searchitoperations.techtarget.com/definition/Nagios>

<https://mindmajix.com/devops-tools-and-frameworks#:~:text=Some%20of%20the%20widely%20used,Scrum%2C%20Sprint%2C%20and%20safe.&text=From%20the%20stage%20to%20development,automation%20of%20the%20delivery%20pipeline.>

<https://es.wikipedia.org/wiki/Jenkins>

<https://riptutorial.com/es/jenkins>

<http://www.robertocrespo.net/kaizen/aprende-a-montar-un-entorno-de-integracion-continua-iv-jenkins/>

<https://www.redhat.com/es/topics/containers/what-is-docker#:~:text=La%20tecnología%20Docker%20usa%20el,puedan%20ejecutarse%20de%20manera%20independiente.&text=Las%20herramientas%20del%20contenedor%2C%20como,de%20implementación%20basado%20en%20imágenes.>

[https://en.wikipedia.org/wiki/Docker_\(software\)#:~:text=Docker%20Inc.,as-a-service%20company.](https://en.wikipedia.org/wiki/Docker_(software)#:~:text=Docker%20Inc.,as-a-service%20company.)

<https://openwebinars.net/blog/3-casos-reales-de-uso-de-docker-en-grandes-empresas/>

<https://tutobasico.com/instalar-docker-windows/>