

Primera Tarea Administración de Datos

Articulo investigativo sobre Docker, Kubernetes y
formatos de guardado de inteligencias artificiales
entrenadas

Estudiante: Gabriel León Castro

Profesor: Rodrigo Herrera

Fecha de entrega: 29/05/2022

En la actualidad es difícil negar que usamos a diario la inteligencia artificial ya sea para el trabajo o entretenimiento y cada vez se involucra más en las vidas cotidianas de las personas para simplificarlas. Esto va desde traductores de idiomas, recomendaciones de contenido o filtros en redes sociales.

Por eso, las tecnologías para la investigación, desarrollo y despliegue de inteligencias artificiales son muy diversas y conforme avanza el tiempo se desarrollan más, pero desde hace unos años hay ciertas tecnologías que han predominado sobre el resto de las otras, por ese motivo en este artículo se profundizará en el despliegue y métodos de guardado de inteligencias artificiales.

Docker

Docker es una tecnología para automatizar la implementación de aplicaciones utilizando contenedores autosuficientes que pueden ser ejecutados localmente o en la nube. Adicionalmente Docker también es una empresa que desarrolla y da soporte a la tecnología previamente explicada.

¿Por qué se crea Docker?

Docker solucionó los problemas al cambiar de entornos de desarrollo principalmente. Este tipo de problemas se dan cuando la aplicación que se está desarrollando requiere una configuración específica, un archivo que no es encontrado en el nuevo entorno, etc.

Aquí es donde entra Docker, ya que con esta herramienta se pueden crear imágenes que contengan lo necesario para que la aplicación pueda ser ejecutada correctamente; desde bibliotecas, código, o dependencias.

¿Cómo funciona?

Docker tiene 2 componentes principales, las imágenes y los contenedores. Las imágenes son una representación estática de la aplicación o el servicio y de su configuración y las dependencias mientras que los contenedores son un contenedor ejecutable, independiente y liviano que incluye todo lo necesario para ejecutar una aplicación incluyendo bibliotecas, herramientas del sistema, código y tiempo de ejecución.

Entonces, Para ejecutar la aplicación o el servicio, se crea una instancia de la imagen de la aplicación para crear un contenedor, que se ejecutará en el host de Docker.

3 ventajas de Docker

1. Elimina las dependencias del sistema y su mantenibilidad se vuelve más simple.
2. Docker al ser portable es simple de implementar en nubes comerciales como Google cloud, AWS o Azure.
3. Capacidad de aislar procesos y recursos de cada contenedor y así no haya conflictos entre contenedores u otros procesos.

3 desventajas de Docker

1. Por si solo se vuelve complicado orquestar muchos contenedores y aplicaciones en contenedores a la vez.
2. No posee una buena implementación en sistemas operativos Windows.
3. Poca compatibilidad con aplicaciones que tengan ventanas gráficas.

Kubernetes

Kubernetes es una plataforma de código abierto portable y extensible para administrar cargas de trabajo, servicios y orquestar funciones entre contenedores de Docker.

¿Por qué se crea Kubernetes?

Kubernetes viene a solucionar el problema que se encuentran los desarrolladores cuando tienen una gran cantidad de contenedores debido a que tiene como objetivo manejar una amplia gama de cargas de trabajo, incluidas aplicaciones con estado y sin estado, así como aplicaciones de procesamiento de datos. En resumen, Kubernetes ayuda a orquestar los contenedores de Docker.

¿Cómo funciona?

Kubernetes se compone por clusters de máquinas y cada una de ellas puede ser Nodo master o Nodo minion.

Los nodos master son los encargados de decidir en qué nodo minion se ejecuta cada contenedor, de mantener el estado del clúster, de asegurar que en todo momento hay el número deseado de contenedores en ejecución y de actualizar las aplicaciones de forma coordinada cuando se despliegan nuevas versiones.

3 ventajas de Kubernetes

1. Kubernetes es una plataforma de autoservicio como PaaS que brinda a los equipos de desarrollo la flexibilidad para abstraer el hardware.
2. Puede ser ejecutado en plataformas en la nube como Google cloud, AWS y Azure.
3. Permite un mejor control de los recursos del cluster.

4 desventajas de Kubernetes

1. Las migraciones se vuelven más complejas.
2. Proceso de instalación y configuración complejo.
3. Tiene incompatibilidad con otras herramientas de Docker.

Formatos de almacenamiento de modelos de inteligencia artificial entrenados

HDF5

Es un paradigma de datos jerárquicos para almacenar y organizar de manera eficiente y flexible volúmenes masivos de datos. Acepta una amplia gama de formatos de datos e incluso te permite crear el tuyo propio. Es posible trabajar con relaciones de datos complejas gracias a las técnicas de agrupación y vinculación.

¿Por qué se crea HDF5?

Se crea con el fin de tener un formato para guardar una amplia cantidad de datos, en especial aquellos conjuntos de datos multidimensionales o datos científicos.

¿Cómo funciona?

La estructura del grupo se puede considerar como una estructura de diccionario anidado en Python, con el directorio raíz que contiene directorios adicionales y el conjunto de datos asociado se almacena en el directorio del nodo. Los datos se pueden estructurar de manera razonable utilizando una estructura jerárquica de este tipo.

3 ventajas de HDF5

1. Se admiten demasiados formatos de interfaz.
2. Admite gran cantidad de datos en un solo archivo.
3. Es mejor que un archivo de texto plano o un XLSX cuando hay varios tipos de datos en un solo conjunto de datos.

3 desventajas de HDF5

1. Su API sea demasiado complicada.
2. El índice de direcciones de 32 bits hace que su límite de almacenamiento sea de solo 2 GB.
3. Las funciones de la biblioteca están desactualizadas.

YAML

YAML es una estructura de datos jerárquica legible por humanos que funciona con cualquier lenguaje informático y se usa más comúnmente para almacenar archivos de configuración.

¿Por qué se crea YAML?

La sintaxis es sencilla y se creó con el objetivo de ser legible y asignable a los tipos de datos más comunes que se encuentran en la mayoría de los lenguajes de alto nivel. Todos los datos se pueden representar de manera efectiva como una combinación de listas, hashes y datos escalares.

¿Cómo funciona?

A diferencia de XML, YAML no tiene etiquetas predefinidas, las mismas propiedades delimitan los datos y aparte se basa en tabulaciones y etiquetas mínimas para definir la información.

3 ventajas de YAML

1. Es de fácil lectura y escritura para los humanos.
2. YAML admite también trabajar con tipos de datos complejos.
3. Un documento YAML es más pequeño que un documento JSON.

3 desventajas de YAML

1. El parser de un documento JSON es más sencillo que el de un documento YAML.
2. En intercambio de datos, la serialización de JSON es más adecuada que YAML.
3. La complejidad del parser de YAML y tareas de serialización no incita a utilizar YAML en sustitución de JSON.

JSON

JSON es un formato de intercambio de datos liviano que es fácil de leer y escribir para los programadores y fácil de interpretar y generar para las máquinas. Sus abreviaturas significan Notación de objetos de JavaScript.

¿Por qué se crea JSON?

Surgió como un remplazo del XML ya que este presentaba problemas sobre todo cuando se trataba de trabajar con gran volumen de datos, puesto que el procesamiento se volvía lento; de esa manera surgiendo el JSON para definir formatos que fueran más ligeros y rápidos para el intercambio de información.

¿Cómo funciona?

Los datos contenidos en un archivo en formato JSON deben estructurarse por medio de una colección de pares con nombre y valor o deben ser una lista ordenada de valores. Sus elementos tienen que contener:

- Clave: corresponde al identificador del contenido. Por eso, debe ser una string delimitada por comillas.
- Valor: representa el contenido correspondiente y puede contener los siguientes tipos de datos: string, array, object, number, boolean o null.

3 ventajas de JSON

1. Es autodescriptivo y fácil de entender.
2. Es más fácil de parsear.
3. Velocidad de procesamiento alta.

3 desventajas de JSON

1. No soporta grandes cargas, solo datos comunes.
2. Para la seguridad requiere de mecanismos externos como expresiones regulares.
3. Algunos desarrolladores encuentran su escueta notación algo confusa.

PLK (Pickle)

Es un módulo Python que permite serializar objetos a archivos en el disco y deserializarlos de nuevo en el programa en tiempo de ejecución. Contiene un flujo de bytes que representa los objetos.

¿Por qué se crea PLK (Pickle)?

Pickle se crea con el fin de almacenar casi cualquier objeto Python directamente en un archivo o cadena sin necesidad de realizar ninguna conversión

¿Cómo funciona?

Lo que el módulo pickle realiza en realidad es lo que se llama serialización de objetos, es decir, convertir objetos a y de cadenas de bytes. El objeto que va a ser pickled se serializará en un flujo de bytes, los cuales se pueden escribir en un archivo, por ejemplo, y restaurar en un punto de ejecución posterior.

3 ventajas de Pickle

1. Es multiplataforma.
2. Puede guardar casi cualquier objeto de Python en un archivo en el disco duro.
3. Es fácil de usar e implementar.

3 desventajas de Pickle

1. Solo está disponible para Python.
2. No tiene una gran comunidad.
3. No tiene una documentación muy detallada.

ML5.js

ML5.js tiene como objetivo hacer que el aprendizaje automático sea más accesible para una amplia audiencia de artistas, diseñadores, programadores creativos y estudiantes. La biblioteca proporciona acceso a algoritmos y modelos de aprendizaje automático en el navegador, basándose en TensorFlow.js sin otras dependencias externas.

¿Por qué se crea ML5.js?

ML5.js se crea con el fin de hacer que el machine learning y diferentes modelos de inteligencias artificiales sean accesibles para la mayoría de la gente sin la necesidad de tener un conocimiento profundo en el tema.

¿Cómo funciona?

Esta pensada para ser ejecutada en el navegador y se usa importando un script de javascript y usando las funciones que trae la librería de ML5.js

3 ventajas de ML5.js

1. Se puede utilizar en el propio navegador sin necesidad de instalar dependencias complicadas.
2. Es fácil de utilizar y de implementar.
3. Trae varios modelos pre-entrenados listos para ser utilizados.

3 desventajas de ML5.js

1. Los modelos no son modificables.
2. Es una librería para usuarios que programan en muy alto nivel.
3. No tiene acceso predeterminado al sistema de archivos en el entorno del host del navegador. Esto limita los recursos de datos disponibles y puede imponer restricciones en el tamaño de los archivos.

P5.js

p5.js es una librería de código abierto escrita en JavaScript que permite crear dibujos y animaciones en el navegador utilizando código de manera simple y accesible para todos los usuarios.

¿Por qué se crea p5.js?

P5.js nace como una librería enfocada en la programación creativa, que busca hacer que programar sea accesible e inclusivo para artistas, diseñadores, educadores, principiantes.

¿Cómo funciona?

Para utilizar p5.js se necesita importar la librería y luego en el código tendrá obligatoriamente dos funciones: *setup* y *draw*. *setup* se invoca una única vez cuando se ejecuta la web, mientras que *draw* se ejecuta dentro de un ciclo hasta que detienes el programa; en esta función es donde se tiene que poner todo el código y métodos que trae la librería para que sea visualizado en la página.

3 ventajas de p5.js

1. Es fácil de utilizar y de implementar.
2. Está pensada para personas que no saben mucho sobre programación por lo que los métodos son de alto nivel y no requieren un amplio expertíz en programación.
3. Tiene una amplia documentación.

3 desventajas de p5.js

1. Al ser de alto nivel no es muy personalizable.
2. No se tiene estandarizado reglas a la hora de programar con esta librería.

Referencias:

“¿Que Es Docker?” Microsoft, 18 May 2022, <https://docs.microsoft.com/es-es/dotnet/architecture/microservices/container-docker-introduction/docker-defined>

¿Que Es YALM?, Red Hat, 18 June 2021, <https://www.redhat.com/es/topics/automation/what-is-yaml>

¿Qué Es Kubernetes?, Kubernetes, 11 Feb. 2022, <https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/>