

ORQUESTACIÓN DE SERVIDORES, KURBENETES, MICROSERVICIOS, OAUTH2.0 E IMPLEMENTACIÓN DE SERVICIOS EN LA NUBE (12-FACTOR APPLICATION)

V. E. Gonzalez Ortiz

7690-13-11075 Universidad Mariano Gálvez

Seminario de Tecnologías de la Investigación

vgonzalezo@miumg.edu.gt

1. Resumen

La implementación de aplicaciones a gran escala han impulsado el uso de arquitecturas flexibles y escalables, se explora las tecnologías mas utilizadas en estas arquitecturas así como los metodos de autenticacion utilizados para la gestion de las aplicaciones y servicios así como la gestion y despliegue.

2. Palabras clave

Orquestacion, gestion, Servicios en la nube, contenedores.

3. Desarrollo del tema

Considerando la implementación y ampliación de la tecnología en gran parte de la vida diaria, como en el trabajo, la educación, la salud, las comunicaciones, entre otras muchas áreas, se deben tener en cuenta las amenazas que esto conlleva, como la integridad de la información que se consulta, la confidencialidad de la información personal o privada, y la accesibilidad o disponibilidad a la información crítica. A raíz de estas amenazas, ahora se tiene un enfoque en poder proteger los datos que la tecnología procesa.

3.1. Temas relacionados

Se detallan a continuacion una serie de subtemas relacionados que apoyan al mejor entendimiento de la informacion de los temas principales, mencionando a su vez la relacion que tienen con los mismos.

Orquestacion

La orquestacion es la coordinacion de forma automatizada de multiples tareas, procesos y servicios.

Application Programing Interface (API) - Interfaz de Programacion de Aplicaciones

Son mecanismos que permiten a dos componentes o servicios de software comunicarse entre ellos por medio de protocolos predefinidos.

Software as a Service (SaaS) - Software como servicio

En este escenario se entrega un software completamente funcional, teniendo la administracion completa al programador, en este aspecto el usuario adquiere el uso de una aplicacion bajo demanda librandose del mantenimiento.

Contenedores

Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de manera confiable en diferentes entornos.

3.2. Orquestación de Servidores

La orquestación de servidores es el proceso automatizado de gestión y coordinación de múltiples servidores. Esta tecnología permite la automatización de tareas como la provisión de recursos, la administración de configuraciones y la implementación de aplicaciones. Las herramientas de orquestación son fundamentales en la administración moderna de infraestructuras IT (tecnología de información) para mantener la coherencia y reducir el error humano en entornos complejos de servidores.

Su funcionalidad

Funciona mayormente para simplificar y agilizar las operaciones en el ambiente de TI permitiendo organizar y gestionar las infraestructuras de manera eficiente.

Como funciona

Se basa mayormente en el uso de herramientas y plataformas que permiten gestionar y ejecutar flujos automatizados, se realiza por medio de un controlador centralizado que tiene la capacidad de coordinar y ejecutar dichos flujos.

3.3. Kurbenetes

Kubernetes es una plataforma portable de código abierto para la automatización, implementación y administración de contenedores, basado en Google Cloud y liberado por Google en el año 2014, utilizada por la popularidad del uso de contenedores en las organizaciones,

Su funcionalidad

Se divide en tres principales funcionalidades o características, **Como plataforma de contenedores**, permite desplegar, escalar y administrar contenedores en un clúster de servidores físicos o virtuales **Plataforma de microservicios, plataforma portable de nube**

3.4. Microservicios

La arquitectura de microservicios es un enfoque para desarrollar aplicaciones como una serie de servicios pequeños e independientes, cada uno enfocado en una funcionalidad específica y se comunica entre ellos a través de APIs definidas. Esta arquitectura favorece la escalabilidad, la flexibilidad y la rapidez en el desarrollo, permitiendo a los equipos trabajar en diferentes partes de una misma aplicación sin interferir entre sí. Sin embargo, la gestión de microservicios a gran escala requiere una orquestación eficiente.

Características

Autonomía: Cada servicio se puede desarrollar, implementar y operar sin afectar el funcionamiento de otros servicios, así como la implementación de tecnologías independientes. **Especializados:** Cada servicio está enfocado en resolver un problema en específico. **Reutilizable:** Al estar bien definida la función de cada servicio, se pueden crear funciones determinadas que pueden ser llamadas y utilizadas en diferentes partes de la aplicación.

3.5. OAuth2.0

Autorización Abierta, es un protocolo de autorización estándar que permite acceso limitado a las aplicaciones a los recursos de otras aplicaciones en nombre de los usuarios sin exponer sus credenciales. la versión 1.0 fue sustituida en el año 2012. Es ampliamente utilizado en la implementación de microservicios y aplicaciones en la nube, permitiendo un control granular sobre los permisos de acceso. Es un protocolo de autorización y no de autenticación, esto quiere decir que es utilizada para conceder acceso a un conjunto de recursos.

Como funciona

Se solicita autorización al servidor de autorización proporcionando el ID la clave secreta de cliente como identificación y los ámbitos a utilizar en el URI de redireccionamiento, posterior el servidor de autorización autentica al cliente y se verifica los ámbitos solicitados están permitidos, una autenticado el servidor redirige de vuelta al cliente con un código de autorización, con el token de acceso, se tiene el uso de los recursos solicitados.

3.6. implementación de servicios en la nube (12-factor application)

La metodología 12-Factor Application es un conjunto de principios para desarrollar aplicaciones modernas que sean escalables, portátiles y fáciles de mantener en entornos en la nube. Este enfoque promueve la separación de configuraciones, la gestión de dependencias, y la ejecución de aplicaciones en contenedores, alineándose a la arquitectura de microservicios. Esta metodología es un marco enfocado en aplicaciones de software como servicio (SaaS) Los doce factores equivalen a un principio y prácticas que se deben de cumplir en las aplicaciones modernas que implementa esta metodología.

1. **Base de Código Única:** utilizar un sistema de control de versiones que permita múltiples despliegues como desarrollo producción y pruebas.
2. **Dependencias:** declarar y aislar las dependencias de aplicaciones en archivos nombrados explícitamente.
3. **Configuración:** Almacenar todas las variables de configuración del entorno en un archivo de configuraciones en vez de tenerlas en el código.
4. **Servicios de Apoyo:** Todo servicio externo como bases de datos y almacenamiento, y servicios externos puedan ser cambiados, actualizados sin necesidad de que el código de la aplicación sea afectado
5. **Construcción, Liberación, Ejecución:** Las etapas deben estar separadas hasta su despliegue
6. **Procesos:** la aplicación funciona como un proceso más, devolviendo un estado y almacenando un respaldo
7. **Enlace de Puertos:** se debe de manejar las solicitudes de HTTP para los servicios WEB a través de un puerto establecido.
8. **Concurrencia:** Ser escalable clonando procesos en vez de utilizar hilos dentro de los procesos.
9. **Disponibilidad:** Se debe de asegurar la solidez de la aplicación y validación de tiempos de ejecución adecuados.
10. **Paridad entre Desarrollo y Producción:** Los entornos deben tener similitud para evitar inconsistencias entre despliegues.

11. **Registros:** Deben de ser registrados y gestionados ppppor separados para la gestion de un analicis.
12. **Procesos de Administración:** Se debe de garantizar la consistencia de las tareas administrativas asi como del entorno.

4. Observaciones y comentarios

1. Estas tecnologías, cuando se implementan de manera conjunta, ofrecen una infraestructura altamente automatizada y escalable que responde a las demandas de las aplicaciones modernas.
2. La complejidad de gestión y la necesidad de habilidades especializadas son desafíos que deben ser abordados cuando se implementa estas metodologías y tecnologías.

5. Conclusiones

1. La adopción de Kubernetes, microservicios, y la metodología 12-Factor Application proporciona una base sólida para la construcción de aplicaciones escalables y resilientes en la nube.
2. OAuth2.0 complementa esta infraestructura al ofrecer un mecanismo seguro y eficiente de autenticación y autorización. Aunque estas tecnologías presentan desafíos, su correcta implementación ofrece beneficios significativos en términos de flexibilidad, escalabilidad y seguridad.

6. Bibliografia

- * Kaspersky. (n.d.). What is cyber security? Kaspersky Latam.. <https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security>
- * Orquestación en servidores - Huawei Enterprise Support Community. <https://forum.huawei.com/enterprise/es/orquestaci%C3%B3n-en-servidores/thread/671615996298936320-667212884846981120>
- * ¿Qué es la orquestación? - Red Hat. <https://www.redhat.com/es/topics/automation/what-is-orchestration>
- * ¿Qué es Kubernetes? - Kubernetes.io. <https://kubernetes.io/es/docs/concepts/overview/what-is-kubernetes/>
- * ¿Qué es Kubernetes? - Google Cloud. <https://cloud.google.com/learn/what-is-kubernetes?hl=es-419>
- * Microservicios - AWS. <https://aws.amazon.com/es/microservices/>
- * ¿Qué es OAuth 2.0? - Auth0. <https://auth0.com/es/intro-to-iam/what-is-oauth-2>
- * 12 Factor App: Aplicaciones Exitosas SaaS - Aplyca. <https://www.aplyca.com/blog/12-factor-app-aplicaciones-exitosas-saas>

https://github.com/UMGVg/SDTDI_Foro_Acad-mico1