



APLICACIÓN WEB PROTOTIPO PARA LA
EVALUACIÓN DE COSTOS Y APROVISIONAMIENTO
CON LIBCLOUD SOBRE AMAZON WEB SERVICES

PRICECLOUD

ANTEPROYECTO - TRABAJO PROFESIONAL

SEBASTIÁN AFANADOR FONTAL

sebastian.afanador@correounivalle.edu.co

Código 1629587

JOHN ALEXANDER SANABRIA ORDÓÑEZ PH.D

john.sanabria@correounivalle.edu.co

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Universidad del Valle
Cali - Colombia

Marzo 2021 – Versión 1.2.0

El Web Scraping son un conjunto de técnicas y tecnologías para extraer información de la web usando algoritmos de exploración recursiva en todos los Endpoints de un sitio web. LibCloud permite generar una capa de abstracción a partir de una API unificada usando controladores para cada tipo de servicio que prestan muchos de los proveedores de Cloud Computing.

RESUMEN

La aplicación web PriceCloud propone integrar la fase de evaluación de costos de los principales proveedores de servicios de cloud computing junto con su despligue en una única interfaz para dar un criterio informado que ayude a minimizar los costos de funcionamiento de un proyecto de software con respecto a sus necesidades iniciales.

Usando principalmente técnicas de *Web Scraping* este trabajo propone crear un servicio que sea una fuente de información veraz y actual de los costos de funcionamiento de los servicios ofrecidos por los principales proveedores de cloud y ademas una aplicación que presente esta información al usuario y le permita desplegar sus recursos usando la librería *LibCloud*¹.

¹ <https://libcloud.apache.org/about.html>

ÍNDICE GENERAL

Acronyms	iii
1 INTRODUCCIÓN	1
2 PLANTEAMIENTO DEL PROBLEMA	2
3 JUSTIFICACIÓN	3
3.1 Justificación Económica	3
3.2 Justificación Académica	3
4 OBJETIVOS	4
4.1 Objetivo general	4
4.2 Objetivos específicos	4
5 RESULTADOS ESPERADOS	5
6 ALCANCES DE LA PROPUESTA	6
7 ANTECEDENTES	7
8 MARCO REFERENCIAL	8
9 METODOLOGÍA	9
10 PRESUPUESTO	10

DEFINICIONES

API

Endpoints

Let'sEncrypt

React

HTTPS

AES

Bots

API REST

LibCloud

aws Amazon Web Services

Azure

Google Cloud

IBM Cloud

DigitalOcean

HostDime

HostingRED

UI

UX

XPath

Docker-compose

INTRODUCCIÓN

El *Cloud Computing* ha facilitado el despliegue de todo tipo de aplicaciones con un uso adaptable de recursos, esto es un punto importante del cloud, permitirle al cliente iniciar con recursos de base que luego pueden ser escalados a la medida de las necesidades que la aplicación requiera. En contraposición de una infraestructura *On-premise* las tecnologías cloud se adaptan en tiempo real a los requerimientos cambiantes de la demanda de estas aplicaciones y le permiten al cliente invertir recursos vitales en procesos mas relevantes eliminando los costos de mantenimiento e infraestructura tecnológica.

El aprovisionamiento de recursos en proveedores de servicios cloud es un proceso que cada vez toma mas relevancia en aplicaciones que han apostado por la tendencia a migrarse al cloud computing. Sin embargo a la hora de elegir estos recursos se deben tener en cuenta aspectos como la arquitectura del proyecto, el tipo de recursos, su ubicación geográfica, las políticas donde recidirán, la reputación del proveedor y por supuesto de su costo.

Uno de los aspectos clave para ayudar a garantizar la continuidad de un proyecto de software son precisamente sus costos fijos; el cloud los discretiza con el modelo *Pay as You Go* en términos de peticiones, duración, poder de cómputo, espacio de almacenamiento, usuarios, y demás variables que le agregan aún mas complejidad a la decisión de escoger el proveedor adecuado para los recursos requeridos por el proyecto.

PriceCloud permitirá interactuar sobre una interfaz web donde el usuario podrá tomar una decisión informada conociendo los costos de su configuración de recursos sobre varios proveedores de cloud y de manera complementaria gracias a los controladores que ofrece *LibCloud* podrá ser asistido en su aprovisionamiento.

El Cloud Computing es una tendencia de servicios y recursos que funcionan sobre Internet para facilitar el despliegue, uso y disponibilidad de aplicaciones de Software.

En una configuración On-premise el cliente se encarga de hospedar sus aplicaciones en uno o varios servidores en su propia infraestructura.

El modelo comercial Pay as You Go permite al cliente pagar solo por lo que ha usado después de haberlo usado.

2

PLANTEAMIENTO DEL PROBLEMA

Según lo mencionado anteriormente esta propuesta de trabajo de grado busca responder las siguientes preguntas. ¿Cómo dar un criterio informado respecto a los costos de aprovisionamiento de una aplicación facilitando su despliegue?, ¿Cómo construir un servicio que brinde información actualizada de los costos de los proveedores?, ¿Cómo proponer un proyecto colaborativo que sea extensible a nuevos proveedores y servicios?, ¿Qué tecnologías y arquitectura usar para procurar su ampliación?.

JUSTIFICACIÓN

3.1 JUSTIFICACIÓN ECONÓMICA

Actualmente no existe una tecnología *Open Souce* que permita administrar recursos en proveedores de Cloud Computing teniendo en cuenta los costos de servicios; gracias a que Price-Cloud será una proyecto de codigo abierto podrá implementarse de manera gratuita por cualquier desarrollador para dar un punto de apoyo en este proceso que carece de un marco bien definido.

3.2 JUSTIFICACIÓN ACADÉMICA

Diseñar e implementar esta propuesta pone a prueba las destrezas técnicas y formación del aspirante a grado; Este proyecto de código libre permitirá tener un punto de partida para que mas interesados puedan contribuir en la solución del problema que parece tener carencia de participación.

La propuesta y desarrollo de este proyecto sugieren conocimientos de Desarrollo de Software, Bases de Datos Relacionales, Aplicaciones Web, Microservicios, Web Scraping, Crawling, Seguridad Web, Web Services, Redes de Computadores y Sistemas Operativos.

4

OBJETIVOS

El lenguaje [XPath](#) es un recurso que permite describir patrones para hallar coincidencias en los [Endpoints](#) de un sitio Web.

La recuperación de información en la web inicia con el [Crawling](#) que permite obtener las páginas web de objetivo para luego aplicar [Web Scraping](#) que recopila información de interés.

4.1 OBJETIVO GENERAL

Diseñar, e implementar un prototipo de aplicación web para informar las tarifas de aprovisionamiento de recursos tipo *Compute*, *Storage* y *Container* en el Cloud Computing y administrar su orquestación.

4.2 OBJETIVOS ESPECÍFICOS

1. Definir un modelo general de evaluación de costos para el análisis de los servicios en el Cloud Computing usando la mayor cantidad de información recuperable y relevante de la Web de sus proveedores.
2. Diseñar e implementar una aplicación basada en microservicios que recopile periódicamente y almacene las tarifas de los principales proveedores de Cloud Computing usando técnicas de *Web Scraping* y *Crawling* con [XPath](#), bases de datos relacionales y una [API REST](#).
3. Diseñar e implementar una aplicación web basada en microservicios para acceder a la información de costos de los proveedores y administrar los recursos de un usuario usando la [API](#) de *LibCloud*.

RESULTADOS ESPERADOS

Cuando se culmine el proyecto propuesto en este escrito se espera cumplir los objetivos específicos mencionados anteriormente como se describe a continuación.

1. Este modelo permitirá tener un punto de referencia para medir los proveedores equitativamente teniendo en cuenta la mayor información que se pueda recuperar de manera pública, se espera poder incluir en todos los casos los costos para cada tipo de servicio, e información adicional como el tiempo en línea y la proximidad geográfica de cada proveedor que permitan tener un criterio mas amplio para tomar una decisión. Este modelo se incorporará al servicio web principal del proyecto.
2. Esta parte del proyecto permitirá manejar de manera centralizada, actualizada y con una trazabilidad la información relevante de los proveedores de cloud, se pretende que sea un servicio en línea que funcione sin interrupciones pues su responsabilidad es responder a las consultas hechas por las instancias de la interfaz web.
3. Esta parte de la aplicación podrá ser desplegada a discreción de cualquier usuario sobre su propia infraestructura para poder usar la solución propuesta en una interfaz Web, estará disponible en una imagen pública en *Dockerhub* con su respectiva documentación y parámetros de configuración usando [Docker-compose](#).

El servicio principal de la aplicación también podrá ser desplegado en un stack de contenedores dispuesto en Dockerhub solo si el usuario está interesado en recuperar por sus propios medios la información de los proveedores.

La comunicación de todos los web services estará encriptada usando el protocolo [HTTPS](#) con certificados libres de [Let'sEncrypt](#), los datos sensibles almacenados usarán el algoritmo de encriptación simétrica AES-256!

6

El archivo robots.txt ubicado en la raíz de un sitio web indica a los Bots que información es de carácter privado o no debe ser indexada.

ALCANCES DE LA PROPUESTA

PriceCloud pretende recuperar de la Web información de los principales proveedores de Cloud entre los que están [aws](#), [Azure](#), [Google Cloud](#), [IBM Cloud](#) y [DigitalOcean](#). PriceCloud también tendrá en cuenta proveedores locales (en Colombia) como [HostDime](#), [HostingRED](#); Se espera que el registro de nuevos proveedores pueda ser hecho por cualquier usuario y validado por el administrador de la aplicación.

PriceCloud tendrá soporte básico de administración para los recursos de tipo *Compute*, *Storage* y *Container* solamente para el proveedor [aws](#) pero con la posibilidad de extender el proyecto a nuevos tipos de servicios teniendo en cuenta las disponibilidad de controladores¹ de la API de *LibCloud*.

La implementación de la interfaz web será construida en [React](#) a partir de la librería *CoreUI*², es la razón por la que esta propuesta no considera el diseño de interfaz de usuario [UI](#) o experiencia de usuario [UX](#) como un componente principal.

La información que se pretende recuperar de la Web de los proveedores de Cloud tendrá en cuenta las limitaciones técnicas y legales que los mismos proveedores incluyan en sus políticas y en el archivo `robots.txt` para evitar incurrir en problemas legales.

La implementación de la aplicación con todos sus componentes tiene como objetivo mostrar la aplicación en funcionamiento, sin embargo los aspectos técnicos adicionales referentes a su despliegue como pruebas de rendimiento, pentesting y mantenimiento no son tenidos en cuenta en esta propuesta.

¹ https://libcloud.readthedocs.io/en/stable/supported_providers.html

² <https://github.com/coreui/coreui-react/>

ANTECEDENTES

8

MARCO REFERENCIAL

METODOLOGÍA

10

PRESUPUESTO
