

# Analisis Perbandingan Serverless Computing Pada Google Cloud Platform

\*Ilmi Barokah<sup>1</sup>, Asriyanik<sup>2</sup>

<sup>1,2</sup> Prodi Teknik Informatika, Universitas Muhammadiyah Sukabumi

**Correspondence Author:** Barokah.ilmi8@gmail.com

**DOI:** <https://doi.org/10.37012/jtik.v7i2.662>

## Abstrak

Serverless computing menjadi paradigma tren baru di cloud komputasi, memungkinkan pengembang untuk fokus pada aplikasi inti logika dan aplikasi prototipe cepat. Serverless computing memiliki biaya yang lebih rendah dan kenyamanan yang diberikannya kepada pengguna yang tidak perlu fokus pada manajemen server. Karena prospeknya yang bagus Serverless computing, dalam beberapa tahun terakhir, sebagian besar vendor cloud utama telah meluncurkan platform komputasi tanpa server komoditas mereka. Namun, karakteristik platform ini belum dipelajari secara sistematis. Perlunya analisis platform ini secara kualitatif yaitu mulai dari pengembangan, penyebaran, dan aspek runtime untuk membentuk taksonomi karakteristik. Google Cloud Platform memiliki beberapa jenis serverless Computing, dalam artikel ini akan dijelaskan mengenai perbandingan antara beberapa Serverless Computing yaitu diantaranya Cloud Functions, App Engine, Cloud Run, dan Google Kubernetes Engine (GKE).

**Kata Kunci :** Cloud Computing, Google Cloud Platform, Serverless Computing

## Abstract

*Serverless computing is becoming a new trend paradigm in cloud computing, allowing developers to focus on core logic applications and rapid prototyping applications. Serverless computing has a lower cost and convenience it provides users who don't need to focus on server management. Due to the good prospects of Serverless computing, in recent years, most of the major cloud vendors have launched their commodity serverless computing platforms. However, the characteristics of these platforms have not been studied systematically. The need for qualitative analysis of this platform, starting from development, deployment, and runtime aspects to form a characteristic taxonomy. Google Cloud Platform has several types of serverless computing, in this article we will explain the comparison between several serverless computing.*

**Keywords:** Cloud Computing, Google Cloud Platform, Serverless Computing

## PENDAHULUAN

Saat ini teknologi mengalami perkembangan menuju arah pencapaian kemudahan dan kenyamanan yang luar biasa seiring dengan berkembangnya jaman. Sehingga memungkinkan berjalannya kegiatan sehari-hari yang dianggap tidak dapat dikerjakan

dalam tempo singkat menjadi sangat mungkin untuk dilakukan dalam waktu singkat. Pengembangan teknologi komputasi berbasis internet saat ini diarahkan pada proses aplikasi sistem yang tidak membutuhkan banyak waktu serta tenaga dan mudah. Namun, ada beberapa permasalahan yang terjadi dalam sistem aplikasi jaringan. Salah satunya yaitu ada suatu perubahan program aplikasi internet pada server jaringan lokal, diantaranya harus diinstal ulang atau disesuaikan lagi, termasuk pada proses penggunaan komputer biasa yang memerlukan sistem operasi dan program aplikasi. Program aplikasi sangat dipengaruhi oleh sistem aplikasi. Misalkan pengguna memilih sistem operasi windows misalnya, maka aplikasinya juga harus berbasis windows. Sistem teknologi informasi cloud computing saat ini sedang hangat dibicarakan. Istilah cloud computing mulai banyak dikenal dan perkembangannya sangat tidak terduga. Teknologi ini digadang-gadang mampu menghilangkan persoalan yang telah dijelaskan sebelumnya. Perusahaan-perusahaan besar di berbagai bidang sekarang memberikan perhatian yang besar ke sana. Cloud computing ini merupakan istilah untuk dunia dan yang sistemnya disewakan. Maksudnya, dalam penggunaannya pelanggan diharuskan untuk menyewa beberapa komponen kerja dan server penyimpanan data sampai data center.

Cloud computing menyediakan layanan yang tercantum di atas untuk bisnis melalui Internet. Menggunakan platform cloud membebaskan profesional TI dari banyak tanggung jawab yang terlibat dalam memelihara layanan ini, memungkinkan mereka untuk fokus pada masalah teknis yang lebih penting. Manfaat lain dari cloud computing terletak pada harga. Bisnis hanya membayar sumber daya cloud yang mereka gunakan, yang dapat menurunkan biaya, membuat infrastruktur mereka lebih efisien, dan memungkinkan layanan untuk diskalakan saat kebutuhan mereka berubah. Mempercepat waktu pengembangan aplikasi dengan menggunakan sumber daya cloud yang dapat ditingkatkan atau diturunkan dengan cepat. Menganalisis data dengan mudah di seluruh bisnis dari satu lokasi di lingkungan cloud yang terhubung Streaming audio dan video tanpa jeda, tidak peduli berapa banyak pengguna yang mengaksesnya Memberikan versi terbaru perangkat lunak sesuai permintaan, di mana pun di dunia. Menyimpan, mencadangkan, dan memulihkan data dengan cepat, mudah, dan tanpa mengkhawatirkan integritas data. Besar atau kecil, bisnis dapat memperoleh manfaat dari Cloud computing dalam beberapa cara. Toko ritel independen mungkin menggunakan cloud untuk layanan pemrosesan pembayaran sederhana, sementara perusahaan global dapat memanfaatkan rangkaian lengkap sumber daya pusat data untuk pengembangan dan

penerapan aplikasi dalam skala besar.

Cloud memungkinkan bisnis untuk mendapatkan sumber daya yang mereka butuhkan tanpa harus memiliki perangkat keras di tempat. Ini berarti bisnis dapat menyebarkan layanan di mana pun dan kapan pun mereka mau. Cloud computing dapat mengurangi biaya dengan memungkinkan perangkat keras dikonsolidasikan dan dikelola di lebih sedikit lokasi. Ini memberikan skalabilitas untuk mendukung beban kerja yang berfluktuasi, dan dapat memberdayakan kolaborasi di antara tim dan lokasi jarak jauh.

Cloud computing menurut definisi google adalah center data yang terletak di luar komputer aplikasi pengguna. Google berkontribusi dalam cloud computing mempunyai tujuan utama untuk memanfaatkan sumber daya komputer yang dimilikinya yang berjumlah sangat besar yang terdiri dari jutaan komputer dan tersebar di seluruh dunia, serta untuk mempercepat pengoperasian yang dilakukan oleh aplikasi-aplikasi web pada saat ini. Cloud computing dalam hal ini yang dikembangkan oleh google pada dasarnya terletak pada sekitaran jaringan komputer utama yang beroperasi untuk aplikasi-aplikasi lainnya seperti Google Search dan Google Mail (gmail). Meskipun demikian, tidak semua aplikasi bisa beroperasi di komputer-komputer milik Google yang membentuk jaringan cloud computing itu. Aplikasi-aplikasi tersebut harus dikembangkan menggunakan aturan-aturan yang telah ditentukan oleh Google agar dapat berjalan dengan baik. Aplikasi seperti itu dinamakan dengan Google App Engine (GAE), pada saat ini aplikasi-aplikasi jenis tersebut dapat ditulis dengan cara yg cukup mudah oleh para pemrogram komputer yang menguasai bahasa pemrograman Java dan Python.

Serverless Computing didefinisikan sebagai model eksekusi cloud yang menyediakan layanan back-end berdasarkan kebutuhan. Vendor cloud secara dinamis menyediakan, mengalokasikan, dan menagih pelanggan hanya untuk sumber daya dan penyimpanan yang mereka gunakan. Dengan cara ini, pelanggan tidak perlu membayar tarif tetap untuk sejumlah server atau bandwidth yang dipesan. Sebagai catatan tambahan bagi mereka yang tidak terbiasa dengan istilah "back-end", ini adalah bagian dari aplikasi yang tidak pernah dilihat pengguna, yang tentu saja mencakup server tempat database pengguna dan file aplikasi berada. Meskipun proses ini disebut "Serverless Computing", server fisik masih merupakan bagian dari persamaan. Namun, dengan model komputasi ini, klien tidak ada hubungannya dengan mereka dan tidak perlu khawatir tentang pemeliharaan dan penyediaan server. Dengan demikian, ini membebaskan pengembang dan pemrogram untuk memfokuskan waktu dan keterampilan komputasi mereka dalam melakukan pekerjaan

mereka daripada mengasuh server. Istilah ini awalnya digunakan untuk menggambarkan situasi back-end-as-a-service, di mana pelanggan menggunakan aplikasi seluler untuk terhubung ke server back-end yang dihosting sepenuhnya di cloud. Namun, istilah tersebut secara bertahap berkembang menjadi solusi fungsi sebagai layanan (walaupun FaaS sebenarnya adalah bagian dari komputasi tanpa server). Klien menulis kode logika bisnis dan mengunggahnya ke penyedia host. Penyedia, pada gilirannya, menangani mesin virtual dan manajemen kontainer, penyediaan perangkat keras, dan tugas terkait seperti multithreading.

Serverless computing memungkinkan pengembang membangun aplikasi lebih cepat dengan menghilangkan kebutuhan mereka untuk mengelola infrastruktur. Dengan aplikasi Serverless, penyedia layanan cloud secara otomatis menyediakan, menskalakan, dan mengelola infrastruktur yang diperlukan untuk menjalankan kode.

Dalam memahami definisi Serverless computing, penting untuk dicatat bahwa server masih menjalankan kode. Nama Serverless berasal dari fakta bahwa tugas yang terkait dengan penyediaan dan pengelolaan infrastruktur tidak terlihat oleh pengembang. Pendekatan ini memungkinkan pengembang untuk meningkatkan fokus mereka pada logika bisnis dan memberikan nilai lebih ke inti bisnis. Serverless computing membantu tim meningkatkan produktivitas mereka dan membawa produk ke pasar lebih cepat dan memungkinkan organisasi untuk lebih mengoptimalkan sumber daya dan tetap fokus pada inovasi

## **METODE PENELITIAN**

Penelitian ini dirancang sebagai model penelitian Deskriptif Kualitatif. Penelitian Deskriptif Kualitatif ditujukan untuk mendeskripsikan dan menggambarkan fenomena-fenomena yang ada, baik bersifat alamiah maupun rekayasa manusia, yang lebih memperhatikan mengenai karakteristik, kualitas, keterkaitan antar kegiatan.

Jenis penelitian deskriptif kualitatif menafsirkan dan menguraikan data yang ada bersamaan dengan situasi yang sedang terjadi. Penelitian ini juga mengungkapkan sikap, pertentangan, hubungan serta penggunaan yang terjadi pada sebuah lingkup responden. Jenis penelitian deskriptif kualitatif menggambarkan kondisi apa adanya, tanpa memberi perlakuan atau manipulasi pada variable yang diteliti. Jenis penelitian deskriptif kualitatif merupakan jenis penelitian dengan proses memperoleh data bersifat apa adanya. Penelitian ini lebih menekankan makna pada hasilnya

## **HASIL DAN PEMBAHASAN**

### **Serverless Computing**

Serverless computing memungkinkan developer fokus pada apa yang paling penting – pengembangan – dan tidak perlu khawatir tentang detail mendasar dari infrastruktur dan pemeliharaan. Manfaat lain seperti penskalaan otomatis, bayar sesuai penggunaan, dan penyediaan sesuai permintaan menjadikan hal ini mudah jika Pengguna membuat aplikasi HTTP stateless, aplikasi web dan seluler, aplikasi IoT dan sensor, sistem pemrosesan data, chatbot, dan banyak lagi. Ada tiga penawaran di Google Cloud Platform yang harus dipertimbangkan, jadi mari selami.

Model computing tradisional bergantung pada mesin virtual atau fisik, di mana setiap instance mencakup sistem operasi yang lengkap, siklus CPU, dan memori. VMware mengkomersialkan ide mesin virtual, dan penyedia cloud menganut konsep yang sama dengan layanan seperti Amazon EC2, Google Compute, dan mesin virtual Azure. Komputasi tanpa server menyediakan pendekatan baru yang menyederhanakan pengelolaan dan mengurangi biaya. Serverless computing adalah model eksekusi berdasarkan permintaan berbasis cloud di mana pelanggan menggunakan sumber daya hanya berdasarkan penggunaan aplikasi mereka. Tidak seperti model mesin virtual tradisional di mana pelanggan harus membangun dan mengelola seluruh VM, komputasi tanpa server menyediakan kemampuan untuk membeli hanya siklus CPU dan memori yang diperlukan untuk mendukung aplikasi menggunakan model bayar per penggunaan berbasis peristiwa. Hal ini memungkinkan tim untuk menghindari banyak biaya dan waktu yang terkait dengan pengelolaan perangkat keras, platform, dan sistem operasi di lokasi, sekaligus mendapatkan fleksibilitas untuk menskalakan dengan cepat dan efisien.

Dalam paradigma ini, dimungkinkan untuk menjalankan seluruh arsitektur tanpa menyentuh server virtual tradisional, baik secara lokal maupun di cloud. Sumber daya tanpa server sangat fleksibel dan disesuaikan berdasarkan aplikasi. REST API, autentikasi, database, email, dan pemrosesan video semuanya memiliki tempat di platform tanpa server. Skala layanan untuk memenuhi permintaan. Tidak perlu merencanakan sumber daya tambahan, memperbarui sistem operasi, atau menginstal kerangka kerja. Penyedia pada dasarnya adalah administrator sistem Pengguna.

Inefisiensi membebani perusahaan teknologi hingga \$100 miliar per tahun. Melakukan

---

pembaruan, menginstal perangkat lunak, dan menyelesaikan masalah perangkat keras membutuhkan waktu pengembang hingga 17 jam setiap minggu. Layanan terkelola yang dihosting cloud menghilangkan tugas kecil sehari-hari yang terkait dengan menghosting infrastruktur TI di tempat. Keamanan, database, dan bahasa pemrograman dengan mudah tetap mutakhir dan aman dalam model tanpa server. AWS Cognito, misalnya, ditagih sebagai layanan autentikasi yang selalu terbaru yang mematuhi spengguna industri yang ketat. Pengembang tetap fokus pada pengembangan teknologi daripada dasar-dasar aplikasi. Tim operasi fokus pada membangun jalur pengiriman yang konsisten alih-alih memperlakukan banyak perangkat keras. Meskipun dihosting secara eksternal, Pengguna masih memiliki kontrol langsung atas detail yang lebih baik seperti bahasa dan versi alat yang digunakan. AWS Lambda memungkinkan pengembang untuk menggunakan NodeJS atau Python sementara Pengguna dapat mengontrol hampir setiap detail dari REST API.

Serverless computing adalah model ekonomi baru untuk komputasi awan yang telah terbukti populer dan telah menjadi keuntungan bagi banyak perusahaan. Menggali kembali ke sejarah penyebaran perangkat lunak, ada server fisik di mana administrator sistem akan mempersiapkan mereka untuk instalasi perangkat lunak. Ini adalah usaha yang rumit dan mahal. Kemudian, pengembang dapat menargetkan server simulasi dan mesin virtual yang meningkatkan fleksibilitas migrasi dan membuat penerapan menjadi lebih mudah. Tindak lanjut dari mesin virtual adalah penerapan kemas, yang menyediakan lingkungan yang ringan menggunakan teknologi seperti Docker, OpenVZ, LXC, zona FreeBSD dan penjara Solaris, dll. Tidak seperti mekanisme penerapan yang disebutkan di atas, tanpa server menjanjikan untuk memberi kita tingkat abstraksi lain . Penyedia tanpa server memungkinkan pengguna untuk menulis dan menyebarkan kode tanpa mengkhawatirkan infrastruktur yang mendasarinya.

Serverless architectures berbasis cloud dan berbasis layanan. Ini berarti, alih-alih menyediakan dan mengelola server mereka sendiri, organisasi membeli layanan komputasi cloud seperti, diantaranya Software as a Service, atau SaaS, memberikan perangkat lunak yang dihosting dan dikelola secara terpusat sesuai permintaan kepada pengguna akhir melalui internet (misalnya, Slack, Jira, Salesforce, HubSpot, dll). Infrastruktur sebagai Layanan, atau IaaS, memberikan sumber daya seperti jaringan, penyimpanan data, dan komputer yang harus dikelola oleh pengguna akhir dan menawarkan sumber daya

komputasi, penyimpanan, dan jaringan penting sesuai permintaan, dengan pembayaran sesuai penggunaan dasar. Platform as a Service, atau PaaS, dibangun di atas IaaS dengan memungkinkan pengguna untuk menyebarkan dan menjalankan aplikasi pada platform terkelola. Berikan semuanya, mulai dari aplikasi sederhana berbasis cloud hingga aplikasi perusahaan canggih yang mendukung cloud. Function as a Service, atau FaaS, adalah jenis PaaS khusus yang memungkinkan pengguna mengembangkan, mengelola, dan menjalankan fungsi aplikasi pada platform terkelola (misalnya, AWS Lambda) tanpa kerumitan dalam membangun dan memelihara infrastruktur. Istilah 'serverless' agak menyesatkan, karena masih ada server yang menyediakan layanan backend ini, tetapi semua ruang server dan masalah infrastruktur termasuk manajemen kapasitas, patching, dan ketersediaan ditangani oleh vendor. Hanya saja perusahaan dapat membangun aplikasi back-end tanpa terlibat langsung dalam memelihara dan mengamankan server. Serverless computing memungkinkan pengembang untuk membeli layanan backend dengan basis 'bayar sesuai penggunaan' yang fleksibel, yang berarti bahwa pengembang hanya perlu membayar untuk layanan yang mereka gunakan. Keamanan arsitektur tanpa server sebagian besar dipastikan oleh penyedia layanan Cloud (CSP). Mereka menangani keamanan komponen komputasi infrastruktur teknologi tanpa server. Inilah sebabnya mengapa teknologi tanpa server dianggap relatif lebih aman daripada model komputasi awan lainnya.

Serverless Environment dapat mengurangi permukaan serangan. Kita hanya perlu fokus pada lapisan aplikasi paling atas dalam model OSI (Open System Interconnection). Memelihara dan menambal celah keamanan ditugaskan ke penyedia cloud. Dilihat dari beberapa tanggung jawab utama penyedia cloud. DIantaranya menambal infrastruktur secara berkala. Mengonfigurasi infrastruktur dengan aman dengan pengaturan keamanan yang tepat untuk melindungi dirinya sendiri. Menyiapkan manajemen akun yang tepat untuk infrastruktur. Pastikan hanya menggunakan sistem operasi dan runtime perangkat lunak yang saat ini didukung. Bertanggung jawab untuk menjaga agar waktu proses perangkat lunak tetap mutakhir dan mengonfigurasinya dengan aman. Mengkonfigurasi database dan konektivitas jaringan dengan aman. Namun, seperti teknologi lain yang ada, teknologi ini tidak kebal terhadap risiko dan ancaman. Ada laporan yang merinci masalah keamanan paling umum dan tantangan yang dihadapi sistem tanpa server saat ini.

## **Manfaat**

Berberapa manfaat yang diperoleh dari layanan serverless computing pada Google Cloud

diantaranya:

1. Kecepatan. Serverless computing memiliki kecepatan dalam fasilitas penerapan dan dapat dioperasikan dalam waktu yang sangat singkat. Dengan demikian, tingkat produktivitas dan fleksibilitas pengguna tidak dibatasi.
2. Pengalaman Pengembang Sederhana. Pengembang dan operator dengan infrastruktur dapat dikelola sepenuhnya secara bebas, tanpa ada hambatan dari penyediaan, konfigurasi, patching, dan pengelola server pengguna.
3. Penskalaan Otomatis. Serverless computing secara otomatis meningkatkan atau menurunkan beban kerja pengguna hingga ke titik paling rendah, tergantung pada kecepatan akses.

### **Fitur Utama**

Tahapan yang umumnya digunakan pada produk serverless computing yaitu membangun aplikasi web yang dapat dibuat berskala dan aman. Membuat kode, membangun, dan menerapkan aplikasi yang dapat diskalakan dalam lingkungan yang terkelola sehingga rencana pengembangan berhasil terlaksana dengan keamanan bawaan, penskalaan otomatis dan manajemen operasional yang cepat. Pembangunan API dapat diskalakan di lingkungan yang di setting agar pengembangan berhasil. Tujuan dari pengembangan REST API ini adalah untuk backend web dan seluler serta mengelola koneksi antara berbagai bagian aplikasi pengguna dan layanan cloud internal dengan API dan layanan internal serverless computing. Merancang aplikasi membutuhkan pertimbangan memproses data. Dalam serverless computing dapat dikelolanya infrastruktur yang sesuai dengan beban kerja pengguna, misalnya dalam hal penskalaan otomatis, otorisasi, serta penyebab terjadinya peristiwa. Penggunaan model komunikasi pub/sub mempermudah untuk penerapan dan pengubahan data dalam skala besar dan meorganisasi data dengan kompleks sehingga menghemat waktu. Produk serverless computing dapat merekam peristiwa dari cloud lain, menangani web book, dan mengelola distribusi peristiwa dan beban kerja ke komponen lain. Sehingga dengan adanya kemampuan ini dapat mempermudah aplikasi pengguna dalam mengatasi kebutuhannya secara kompleks. Serverless computing mampu secara otomatis untuk memvalidasi kebijakan atau konfigurasi dan melakukan otomatisasi skrip lainnya.

Platform Serverless compute seperti Cloud Functions, App Engine, dan Cloud Run memungkinkan pengguna dapat membangun, mengembangkan, dan menerapkan aplikasi sekaligus menyederhanakan pengalaman developer dengan menghilangkan semua

pengelolaan infrastruktur. Di sisi lain, Google Kubernetes Engine (GKE) menjalankan Kubernetes Bersertifikat yang membantu pengguna memfasilitasi orkestrasi container melalui konfigurasi deklaratif dan otomatisasi. Platform Serverless Google dan GKE memungkinkan pengguna menskalakan aplikasi berdasarkan kebutuhan infrastruktur. Berikut tabel untuk membantu mengidentifikasi kapan harus menggunakan layanan khusus ini.

### **Cloud Functions**

Solusi berbasis peristiwa yang diperluas ke Google dan layanan pihak ketiga sangat cocok untuk fungsi cloud, serta yang perlu diskalakan dengan cepat. Meskipun App Engine mendukung banyak layanan berbeda dalam satu aplikasi, Cloud Functions mendukung layanan individual. Jika persyaratan Pengguna tidak mencakup beberapa layanan, maka Cloud Functions adalah pilihan yang tepat. Misalnya jika Pengguna menyimpan atau mengekstrak data dari database, memposting file, atau melakukan validasi data sederhana, maka menggunakan Cloud Functions adalah pilihan yang tepat. Fungsi beroperasi di lingkungan runtime mereka sendiri dan berjalan secara independen; ketika suatu fungsi dipanggil, ia berjalan dalam instance terpisah dari panggilan fungsi lainnya. Tidak seperti di App Engine di mana data dibagikan di antara instance, setelah fungsi dipanggil dengan Cloud Functions, fungsi itu sendiri, jadi jika Pengguna perlu melacak data saat menggunakan Cloud Functions, Pengguna harus menggunakan database atau yang dapat ditulis file di Penyimpanan Cloud. Saat ini GCP mendukung Node.js, Python, dan Go.

Meskipun Cloud Run mengambil container dan membuatnya dapat dipanggil melalui permintaan HTTP, Cloud Functions tetap sebagai platform serverless berbasis peristiwa milik Google. Berbeda dengan dikemas dalam wadah Docker, Pengguna menerapkan kode sebagai fungsi. Google mendukung penulisan Cloud Functions sehingga dapat juga dipanggil melalui permintaan HTTP, atau disetel ke pemicu berdasarkan peristiwa latar belakang. Cloud Functions memberikan lebih banyak batasan tentang cara penerapan kode Pengguna (yang jelas adalah Pengguna perlu mengemasnya sebagai suatu fungsi), dan hanya mendukung kumpulan bahasa tertentu (Pengguna dapat menulis Cloud Functions menggunakan JavaScript/Node.js, Python 3, atau Go runtimes), tetapi menawarkan kemampuan untuk memicu fungsi menggunakan peristiwa di lingkungan cloud Pengguna.

Cloud Functions adalah platform Serverless yang terkelola sepenuhnya untuk membuat fungsi mandiri yang merespons peristiwa waktu nyata tanpa perlu mengelola server,

mengonfigurasi perangkat lunak, memperbaiki kerangka kerja, dan menambal sistem operasi. Dengan Cloud Functions, pengguna dapat menulis fungsi tujuan tunggal sederhana yang dilampirkan ke peristiwa yang dihasilkan dari infrastruktur dan layanan cloud. Cloud Functions dapat ditulis menggunakan JavaScript, Python 3, Go, atau runtime Java yang membuat portabilitas dan pengujian lokal lebih familiar. Lingkungan eksekusi sering diinisialisasi dari awal, yang disebut cold start dan membutuhkan banyak waktu untuk menyelesaikannya. Ini adalah lingkungan eksekusi Serverless yang dapat digunakan untuk membangun dan menghubungkan layanan cloud. Ini dapat melayani beban kerja IoT, ETL, webhook, pesan Kafka, analitik, dan layanan berbasis peristiwa. Cloud Functions sangat bagus untuk membangun backend tanpa server, melakukan pemrosesan data waktu nyata, dan membuat aplikasi AI.

### **Fitur utama Cloud Functions**

1. Pengalaman pengembang yang disederhanakan dan peningkatan kecepatan pengembang. Cloud Functions memiliki pengalaman pengembang yang sederhana dan intuitif. Cukup tulis kode Pengguna dan biarkan Google Cloud menangani infrastruktur operasional. Kembangkan lebih cepat dengan menulis dan menjalankan cuplikan kode kecil yang merespons peristiwa. Terhubung ke Google Cloud atau layanan cloud pihak ketiga melalui pemicu untuk menyederhanakan masalah orkestrasi yang menantang.
2. Bayar hanya untuk apa yang Pengguna gunakan. Pengguna hanya ditagih untuk waktu eksekusi fungsi Pengguna, diukur ke 100 milidetik terdekat. Pengguna tidak membayar apa pun saat fungsi Pengguna menganggur. Cloud Functions otomatis berputar ke atas dan ke bawah sebagai respons terhadap peristiwa.
3. Hindari penguncian dengan teknologi terbuka. Gunakan kerangka kerja FaaS (fungsi sebagai layanan) sumber terbuka untuk menjalankan fungsi di berbagai lingkungan dan mencegah penguncian. Lingkungan yang didukung mencakup Cloud Functions, lingkungan pengembangan lokal, lokal, Cloud Run, dan lingkungan tanpa server berbasis Knative lainnya.

### **App Engine**

Google App Engine adalah platform tanpa server yang dikelola sepenuhnya untuk mengembangkan dan menghosting aplikasi web dalam skala besar. Pengguna dapat memilih

dari beberapa bahasa, pustaka, dan kerangka kerja populer untuk mengembangkan aplikasi mereka, lalu App Engine menangani penyediaan server dan penskalaan instance aplikasi berdasarkan permintaan. Ini adalah PaaS untuk membangun aplikasi yang skalabel. Google Cloud menyediakan 2 lingkungan untuk menggunakan App Engine, salah satunya adalah lingkungan standar dengan lingkungan terbatas dan dukungan untuk bahasa seperti Python, Go, node.js. yang lainnya adalah lingkungan Fleksibel di mana pengembang memiliki lebih banyak kebebasan seperti menjalankan runtime khusus menggunakan buruh pelabuhan, waktu tunggu permintaan & respons yang lebih lama, dan kemampuan untuk menginstal dependensi/perangkat lunak khusus, dan SSH ke dalam mesin virtual.

Standard Environment, Ini didasarkan pada wadah yang berjalan di infrastruktur Google. Ini memberi pengguna fasilitas untuk dengan mudah membangun dan menyebarkan aplikasi yang berjalan di bawah beban berat dan sejumlah besar data. Ini mendukung bahasa berikut: Python, JAVA, Node.js, Ruby, PHP, dan Go. Flexible Environment, App Engine memungkinkan pengguna berkonsentrasi pada penulisan kode. Berdasarkan Google Compute Engine, secara otomatis menskalakan aplikasi ke atas dan ke bawah dan bersamaan dengan itu juga menyeimbangkan beban. Ini memungkinkan pengguna untuk menyesuaikan runtime mereka dan sistem operasi mesin virtual mereka menggunakan Dockerfiles.

App Engine adalah platform aplikasi serverless yang dikelola sepenuhnya oleh Google untuk web dan backend API. Meskipun fungsi tanpa server memungkinkan Pengguna menjalankan fungsionalitas yang ringan dan mandiri dengan mudah, mungkin sulit untuk menjalankan aplikasi yang lebih kompleks menggunakan Cloud Functions. Untuk pengembang yang ingin membangun aplikasi tanpa server dengan banyak fungsi, atau mempertahankan beberapa tingkat konteks yang bertahan melampaui permintaan individual, Google App Engine menghadirkan opsi yang menarik. Di Google App Engine, Pengguna cukup mengambil kode dan menerapkannya di Google, lalu membayar sumber daya yang Pengguna gunakan – ini berjalan di App Engine sebagai satu sumber daya yang terdiri dari satu atau beberapa layanan. Untuk setiap layanan, Pengguna menerapkan satu atau beberapa versi layanan tersebut, yang pada gilirannya berjalan dalam satu atau beberapa instance, bergantung pada jumlah lalu lintas yang ditangani setiap versi.

Membangun aplikasi Serverless dengan backend web atau api dan mendukung beberapa bahasa pengembangan tanpa perlu khawatir tentang dukungan infrastruktur. Saat Pengguna memiliki aplikasi yang perlu berkomunikasi dengan beberapa layanan seperti aplikasi web

atau API, App Engine adalah solusi yang sesuai. Hubungan antara layanan tersebut berarti aplikasi diperlakukan sebagai entitas yang dikelola.

App Engine adalah platform Serverless yang dikelola sepenuhnya untuk menghosting dan mengembangkan aplikasi web yang sangat skalabel. Ini memungkinkan pengguna fokus pada kodenya sementara App Engine mengelola masalah infrastruktur. Pengguna dapat menskalakan aplikasinya dari nol hingga skala planet tanpa harus khawatir dan mengelola infrastruktur. Pengguna dapat membangun aplikasinya di Node.js, Java, Ruby, C#, Go, Python, atau PHP runtimes. Selain itu, Pengguna juga dapat membawa pustaka dan kerangka kerja apa pun ke App Engine dengan menyediakan Docker container. Setiap proyek Cloud hanya dapat berisi satu aplikasi App Engine. Setelah App Engine dibuat pada sebuah proyek, Pengguna tidak diizinkan untuk mengubah lokasi aplikasinya. App Engine dapat dengan mulus menghosting berbagai versi aplikasi, dan membantu pengguna dengan mudah membuat lingkungan pengembangan, pengujian, staging, dan produksi. Dengan App Engine, Pengguna dapat merutekan lalu lintas masuk ke berbagai versi aplikasinya, mengujinya secara A/B, dan melakukan peluncuran fitur tambahan dengan menggunakan pemisahan lalu lintas. App Engine terintegrasi dengan mudah dengan Cloud Monitoring dan Cloud Logging untuk memantau kesehatan dan kinerja aplikasi. Ini juga berfungsi dengan Cloud Debugger dan Pelaporan Kesalahan untuk membantu pengguna mendiagnosis dan memperbaiki bug dengan cepat. Pengguna dapat menjalankan aplikasi di App Engine menggunakan lingkungan stpenggunar atau fleksibel. Pengguna diizinkan untuk menggunakan kedua lingkungan secara bersamaan untuk aplikasi guna memanfaatkan manfaat individu masing-masing lingkungan.

### **Manfaat App Engine**

1. Bahasa dan alat yang terbuka dan familiar. Bangun dan terapkan aplikasi dengan cepat menggunakan bahasa populer atau bawa runtime dan kerangka kerja bahasa Pengguna sendiri. Pengguna juga dapat mengelola sumber daya dari baris perintah, men-debug kode sumber, dan menjalankan kembali API dengan mudah.
2. Cukup tambahkan kode. Fokus pada penulisan kode tanpa harus mengelola infrastruktur yang mendasarinya. Lindungi aplikasi Pengguna dari ancaman keamanan menggunakan kemampuan firewall, aturan IAM, dan sertifikat SSL/TLS terkelola.
3. Bayar hanya untuk apa yang Pengguna gunakan. Beroperasi di lingkungan tanpa server tanpa khawatir tentang penyediaan yang berlebihan atau kurang. App Engine secara

otomatis menskalakan bergantung pada lalu lintas aplikasi Pengguna dan menggunakan sumber daya hanya saat kode Pengguna berjalan.

### **Fitur utama App Engine**

1. Bahasa pemrograman populer. Bangun aplikasi Pengguna di Node.js, Java, Ruby, C#, Go, Python, atau PHP—atau bawa runtime bahasa Pengguna sendiri.
2. Terbuka dan fleksibel. Waktu proses khusus memungkinkan Pengguna membawa pustaka dan kerangka kerja apa pun ke App Engine dengan menyediakan wadah Docker.
3. Dikelola sepenuhnya. Lingkungan yang terkelola sepenuhnya memungkinkan Pengguna fokus pada kode sementara App Engine mengelola masalah infrastruktur.

### **Cloud Run**

Cloud Run adalah platform Serverless computing yang sepenuhnya dikelola oleh Google, jadi Pengguna tidak perlu khawatir tentang dukungan aplikasi backend. Tidak seperti App Engine dan Cloud Functions, Cloud Run dapat menskalakan container stateless dan memanfaatkan Google Kubernetes Engine. Jika Pengguna memerlukan opsi Serverless yang memerlukan aplikasi untuk dijalankan dalam container stateless, Cloud Run mungkin merupakan pilihan terbaik untuk penerapan semacam ini. Ini sepenuhnya dikelola, dan harga hanya didasarkan pada sumber daya yang dikonsumsi. Selain itu, jika Pengguna memiliki arsitektur campuran dengan resource lokal dan cloud, Anthos, opsi container tanpa server GKE (Google Kubernetes Engine) tersedia untuk Cloud Run, jadi ini adalah pilihan tepat untuk skenario ini.

Cloud Run adalah lingkungan komputasi yang terkelola sepenuhnya untuk menerapkan dan menskalakan container HTTP tanpa server tanpa khawatir tentang penyediaan mesin, mengonfigurasi cluster, atau penskalaan otomatis. Tidak ada vendor lock-in - Karena Cloud Run menggunakan container OCI standar dan mengimplementasikan Knative Serving API standar, Anda dapat dengan mudah mem-porting aplikasi Anda ke lokal atau lingkungan cloud lainnya. Penskalaan otomatis cepat - Layanan mikro yang diterapkan dalam skala Cloud Run secara otomatis berdasarkan jumlah permintaan yang masuk, tanpa Anda harus mengonfigurasi atau mengelola kluster Kubernetes yang lengkap. Cloud Run menskalakan ke nol—yaitu, tidak menggunakan sumber daya—jika tidak ada permintaan. Membagi lalu lintas - Cloud Run memungkinkan Anda membagi lalu lintas di antara beberapa revisi, sehingga Anda dapat melakukan peluncuran bertahap seperti penerapan canary atau

penerapan biru/hijau. Domain kustom - Anda dapat menyiapkan pemetaan domain kustom di Cloud Run dan ini akan menyediakan sertifikat TLS untuk domain Anda. Redundansi otomatis - Cloud Run menawarkan redundansi otomatis sehingga Anda tidak perlu khawatir membuat banyak instans untuk ketersediaan tinggi.

Dibuat dari Knative, Cloud Run adalah yang terbaru dari penawaran tanpa server Google. Sementara platform tanpa server lainnya menggunakan fungsi yang digerakkan oleh peristiwa sebagai unit utama penerapan, Cloud Run memungkinkan Pengguna untuk mengemas kode dalam wadah tanpa status, lalu memanggilnya melalui permintaan HTTP. Men-deploy container Cloud Run di lingkungan Google yang terkelola sepenuhnya memberi developer manfaat biasa tanpa server (tanpa pengelolaan infrastruktur, penetapan harga berbasis penggunaan, penskalaan otomatis yang lebih mudah), tetapi juga mendukung sejumlah bahasa pemrograman, library, atau binari sistem. Cloud Run juga dapat men-deploy container di Google Kubernetes Engine (GKE), dengan kemampuan untuk secara khusus mengonfigurasi persyaratan hardware untuk container tanpa server Pengguna dalam kasus terakhir. Dengan fleksibilitas ini, pengguna Cloud Run dapat dengan mudah menjalankan beban kerja tanpa server dengan alat yang telah mereka gunakan untuk mengemas dan menjalankan container di Google Cloud, atau menerapkan beban kerja stateful dan stateless secara bersamaan.

Cloud Run adalah platform komputasi tanpa server terkelola yang membantu pengguna menjalankan aplikasi dalam container yang sangat skalabel yang dapat dipanggil melalui permintaan web atau peristiwa Pub/Sub. Dibangun di atas Knative stpenggunar terbuka, yang memungkinkan portabilitas aplikasi pengguna. pengguna dapat memilih bahasa pemrograman pilihan, perpustakaan sistem operasi apa pun, atau bahkan membawa binari pengguna sendiri. Pengguna dapat memanfaatkan alur kerja container karena Cloud Run terintegrasi dengan baik dengan layanan di ekosistem container seperti Cloud Build, Artifact Registry, Docker. Instance container pengguna berjalan di lingkungan sandbox aman yang diisolasi dari sumber daya lain. Dengan Cloud Run, pengguna dapat menaikkan atau menurunkan skala secara otomatis dari nol ke N bergantung pada lalu lintas. Layanan Cloud Run bersifat regional dan secara otomatis direplikasi di beberapa zona. Cloud Run menyediakan integrasi siap pakai dengan Cloud Monitoring, Cloud Logging, Cloud Trace, dan Error Reporting untuk memantau performa kesehatan aplikasi.

### **Manfaat Cloud Run**

Berberapa manfaat yang diperoleh dari layanan Google Cloud Run diantaranya:

1. Wadah untuk produksi dalam hitungan detik. Tulis kode sesuai keinginan Pengguna dengan menerapkan kode atau wadah apa pun yang mendengarkan permintaan atau acara. Bangun aplikasi dalam bahasa favorit Pengguna, dengan dependensi dan alat favorit Pengguna, dan terapkan dalam hitungan detik.
2. Dikelola sepenuhnya. Cloud Run mengabstraksikan semua manajemen infrastruktur dengan menaikkan dan menurunkan skala secara otomatis dari nol hampir secara instan—bergantung pada lalu lintas. Cloud Run hanya menagih Pengguna untuk sumber daya yang tepat yang Pengguna gunakan.
3. Pengalaman pengembang yang ditingkatkan. Cloud Run membuat pengembangan dan penerapan aplikasi menjadi lebih sederhana dan lebih cepat. Dan itu sepenuhnya terintegrasi dengan Cloud Code, Cloud Build, Cloud Monitoring, dan Cloud Logging untuk pengalaman pengembang ujung-ke-ujung yang ditingkatkan

### **Google Kubernetes Engine (GKE)**

Google Kubernetes Engine (GKE) adalah layanan Kubernetes terkelola yang memfasilitasi orkestrasi container melalui konfigurasi dan otomatisasi deklaratif. Ini terintegrasi dengan Identity Access Management (IAM) untuk mengontrol akses di cluster dengan akun Google pengguna dan izin peran yang pengguna tetapkan. GKE menjalankan Kubernetes Bersertifikat. Hal ini memungkinkan portabilitas ke platform Kubernetes lain di seluruh cloud dan beban kerja lokal. Pengguna dapat menghilangkan biaya overhead operasional dengan mengaktifkan perbaikan otomatis, peningkatan otomatis, dan saluran rilis. GKE memungkinkan pengguna memesan rentang CIDR untuk clusternya, memungkinkan IP cluster pengguna untuk hidup berdampingan dengan IP jaringan pribadi melalui Google Cloud VPN. Dengan GKE, pengguna dapat memilih cluster yang dirancang untuk ketersediaan, stabilitas versi, isolasi, dan persyaratan traffic pod dari beban kerja penting. Pengguna dapat secara otomatis menskalakan penerapan aplikasi pengguna ke atas dan ke bawah berdasarkan penggunaan CPU dan memori. Secara default, node cluster pengguna secara otomatis diperbarui dengan versi rilis terbaru Kubernetes. Update rilis Kubernetes tersedia dengan cepat di dalam GKE. Google Kubernetes Engine terintegrasi dengan baik dengan Cloud Logging dan Cloud Monitoring melalui Cloud Console, sehingga memudahkan untuk mendapatkan wawasan tentang aplikasi pengguna.

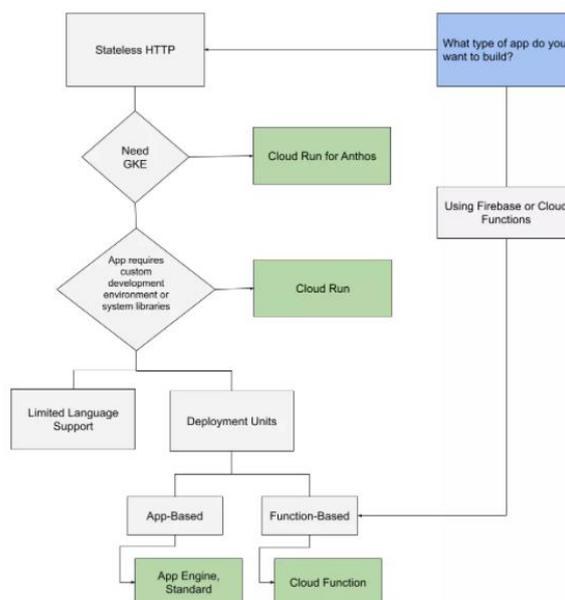


Figure 1. Memilih antara App Engine, Cloud Functions, dan Cloud Run

Ada sembilan area yang perlu Pengguna perhatikan untuk memutuskan layanan GCP mana yang harus Pengguna gunakan, jadi mari kita jelajahi ini lebih detail:

1. Deployment Type, Metode pengiriman untuk menyebarkan aplikasi stateless
2. Scaling, Identifikasi apakah penskalaan memungkinkan dan jika memungkinkan, metode penskalaan apa yang diaktifkan. Cloud Run for Anthos diskalakan secara berbeda dari aplikasi stpenggunar.
3. Runtime environment support, Dukungan bahasa bervariasi antar layanan di GCP. Ragam Cloud Run dan Cloud Run/Anthos mendukung hampir semua bahasa umum, tetapi App Engine dan Cloud Functions mendukung dukungan bahasa yang lebih terbatas.
4. Access controls, Ada pendekatan berbeda untuk menangani keamanan dan manajemen identitas untuk aplikasi stateless di GCP dan masing-masing menangani hal-hal yang sedikit berbeda.
5. HTTP/gRPC, Hanya Cloud Run dengan Anthos yang menangani pengelolaan Cloud Endpoints karena persyaratan topologi campuran
6. Custom domains, Aplikasi mandiri dan berbasis container mendukung pemetaan nama domain kustom untuk aplikasi Pengguna, tetapi platform Cloud Functions tidak.

7. GPU/TPU support, Kemampuan GPU yang diperluas menggunakan unit pemrosesan grafis (GPU) atau unit pemrosesan tensor (TPU) hanya didukung dengan Cloud Run for Anthos.
8. Timeout (minutes), Jumlah menit sebelum batas waktu tanpa aktivitas berbeda antara jenis penerapan stateless. Rasa Cloud Run keduanya menawarkan jendela waktu tunggu 15 menit; Waktu Cloud Functions habis pada 9 menit, dan App Engine pada 1 menit.
9. VPC Networking, Semua aplikasi stateless kecuali Cloud Run memungkinkan jaringan VPC. Pada saat penulisan ini, GCP tidak memiliki jaringan VPC dalam ketersediaan umum.

### Perbandingan Tipe Aplikasi Stateless

Tabel 1. Perbandingan Tipe Aplikasi Stateless

	App Engine	Cloud Functions	Cloud Run
<b>Deployment Type</b>	App	Function	Container
<b>Scaling</b>	Yes	Yes	Yes
<b>Runtime support</b>	Java Node.js Python Go PHP	Node.js Python Go	All available
<b>Access controls</b>	Auth 2.0 CICP Firebase auth Google Sign-in Users API	IAM permissions	IAM permissions CICP Google Sign-in Firebase auth
<b>HTTP/gRPC</b>	No	No	No
<b>Custom domains</b>	Yes	No	Yes
<b>GPU/TPU support</b>	No	No	No
<b>Timeout (mins)</b>	1	9	15
<b>VPC networking</b>	Yes	Yes	No

Dari segi tipe deployment, App engine berjenis tipe App, Cloud Functions berjenis tipe Function dan Cloud Run berjenis tipe Container. Ketiga-tiganya memiliki scaling. Runtime support untuk App engine adalah Java, Node.js, Python, Go dan PHP. Runtime support untuk Cloud Functions adalah Node.js, Python, dan Go. dan untuk Cloud Run Runtime support kesemuanya. Access controls App engine adalah Auth 2.0, CICP, Firebase auth, Google Sign-in dan Uses API. Access controls Cloud Functions adalah IAM permissions. Dan access controls Cloud Run adalah AIM permissions, CICP, Firebase auth, Google Sign-in. Ketiga-

tinganya tidak support HTTP/gRPC. App Engine dan Cloude Run support terhadap Costom domain, sedangkan untuk Cloud Functions tidak support. Timeout (mins) App engine adalah 1, Timeout (mins) Cloud Functions adalah 9, dan Timeout (mins) Cloud Run adalah 15. App engine dan Cloud functions support terhadap VPC networking, sedangkan Cloud Run tidak support VPC networking

## **SIMPULAN**

Dalam artikel ini, kami telah mengkarakterisasi empat komoditas utama platform serverless computing (yaitu, Google Cloud Functions, Google App Engine, Google Cloud). Secara analisis kualitatif, masing masing memiliki kelebihan dan kekurangan sendiri sendiri. Ada karakteristik yang berbeda untuk setiap pendekatan, dan pilihan terbaik untuk Pengguna bergantung pada kebutuhan aplikasi dan preferensi pengembangan Pengguna.

Jika Pengguna sudah mengemas kode dalam container Docker atau menjalankan cluster Kubernetes di Google Cloud, pertimbangkan Cloud Run atau Knative untuk beban kerja tanpa server Pengguna. Untuk menjalankan kode yang merespons peristiwa real-time, atau untuk melayani permintaan tanpa container, gunakan Cloud Functions. Untuk saat Pengguna membutuhkan beberapa bagian fungsionalitas di satu tempat dan hanya ingin menerapkan seluruh aplikasi Pengguna, lihat App Engine.

## **REFERENSI**

- Adzic, G., & Chatley, R. (2017, August). Serverless computing: economic and architectural impact. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering* (pp. 884-889).
- Kelly, D., Glavin, F., & Barrett, E. (2020, October). Serverless Computing: Behind the Scenes of Major Platforms. In *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)* (pp. 304-312). IEEE.
- Kim, J., & Lee, K. (2019, July). Functionbench: A suite of workloads for serverless cloud function service. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)* (pp. 502-504). IEEE.
- Landoni, M., Taffoni, G., Bignamini, A., & Smareglia, R. (2019). Application of google cloud platform in astrophysics. arXiv preprint arXiv:1903.03337.

- Li, J., Kulkarni, S. G., Ramakrishnan, K. K., & Li, D. (2019, December). Understanding open source serverless platforms: Design considerations and performance. In Proceedings of the 5th International Workshop on Serverless Computing (pp. 37-42).
- Lloyd, W., Ramesh, S., Chinthalapati, S., Ly, L., & Pallickara, S. (2018, April). Serverless computing: An investigation of factors influencing microservice performance. In 2018 IEEE International Conference on Cloud Engineering (IC2E) (pp. 159-169). IEEE.
- Lynn, T., Rosati, P., Lejeune, A., & Emeakaroha, V. (2017, December). A preliminary review of enterprise serverless cloud computing (function-as-a-service) platforms. In 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (pp. 162-169). IEEE.
- Malawski, M., Gajek, A., Zima, A., Balis, B., & Figiela, K. (2020). Serverless execution of scientific workflows: Experiments with hyperflow, aws lambda and google cloud functions. *Future Generation Computer Systems*, 110, 502-514.
- Mohanty, S. K., Premsankar, G., & Di Francesco, M. (2018, December). An Evaluation of Open Source Serverless Computing Frameworks. In CloudCom (pp. 115-120).
- Naranjo, D. M., Risco, S., de Alfonso, C., Pérez, A., Blanquer, I., & Moltó, G. (2020). Accelerated serverless computing based on GPU virtualization. *Journal of Parallel and Distributed Computing*, 139, 32-42.
- Wen, J., Liu, Y., Chen, Z., Ma, Y., Wang, H., & Liu, X. (2020). Understanding Characteristics of Commodity Serverless Computing Platforms. arXiv preprint arXiv:2012.00992.
- Google, Serverless computing,. [Online]. Available: <https://cloud.google.com/serverless>. [Accessed Agust. 08, 2021].
- S. M. Shoaib, Google Cloud Functions vs App Engine vs Cloud Run vs GKE,. [Online]. Available: <https://cloud.google.com/serverless>. [Accessed Agust. 08, 2021].
- Splunk, Comparing Google's Serverless Offerings: Cloud Run, Cloud Functions, App Engine,. [Online]. Available: [https://www.splunk.com/en\\_us/blog/devops/gcp-serverless-comparison.html](https://www.splunk.com/en_us/blog/devops/gcp-serverless-comparison.html). [Accessed Agust. 08, 2021].
- Sphere Partners, When to Choose App Engine vs Cloud Functions or Cloud Run in GCP,. [Online]. Available: <https://www.sphereinc.com/blogs/when-to-choose-app-engine-vs-cloud-functions-or-cloud-run-in-gcp/>. [Accessed Agust. 08, 2021].