A black and white logo

Description automatically generated with medium confidence

**Cloud research document**

OFS Platform

By Mohammed Al Harbi

Contents

[Introduction 3](#_Toc121250789)

[Why cloud services exist? 3](#_Toc121250790)

[What are the available cloud services 3](#_Toc121250791)

[Which cloud service is suitable for OFS? 3](#_Toc121250792)

[Function as a service 3](#_Toc121250793)

[Infrastructure as a service 3](#_Toc121250794)

[Services cost analysis 3](#_Toc121250795)

[Conclusion 3](#_Toc121250796)

# Introduction

In this document, we will shift back in time to understand how life was without cloud services and why we need them today. Afterwards we will analyze what the available cloud services are and determine which vendor aligns with Oman freelancing platform application. Then, we will have a look at important concepts of functions as a service and infrastructure as a service. Lastly, we will have a look at the architectural design choice choices in terms services in the selected vendor to have a clear view over cost per service.

# Why cloud services exist?

Going back in time, before 1960’s businesses needed to build software applications and manage their own services. This including all kind of maintaining the physical servers, patching those servers, installing operating systems and provisioning them constantly. The effort required for a simple software was quite a lot. Software programming languages and dependencies had huge impact on business as this will be installed manually on servers, so the focus on business logic was very little and this explains we had such cumbersome functionality back in time. In addition to businesses having to manage their own servers, scalability had to be defined initially and it had fixed costs whether the servers are fully utilized or not; this was very inflexible for businesses and thus needed experts to estimate the number of servers based on business use case. Growing businesses was a disadvantage in terms of software which should not be case, as this means they had to shift at some point from monolithic architectural solution to microservices otherwise they would face monolithic hell at some as they keep growing and increase the size of development team. In case businesses back then switched to microservices, growing businesses on one hand is solved, on the other hand new challenges are introduced.

1. Complexity; as each service is essentially a server that needed to be managed
2. Networking; separating concerns but on servers level require servers to communicate to perform some business logic functionalities and thus needed to be carefully design to work synchronously and asynchronously.
3. Inflexbility; when the design was monolithic businesses just scale vertically by increasing the number of servers their core solution built on but since microservices scale horizontally by increasing the number of micro-servers that essentially perform one business idea and not the complete solution, businesses needed to predict the number of servers per businesses idea (must predict where most load at prior shift to microservices.)

After some time, the problem of having to deal with the programming language and dependences of the built solution have been facilitated with the introduction of Docker, and with the introduction of Kubernetes businesses were encouraged to shift to microservices architecture as some of the problems have been tackled.

In 2006 the introduction of cloud computing was a new technology era. The cloud computing technology aimed to solve common present core software businesses face including managing, patching, installing operating systems, maintaining physical conditions of servers. Nowadays cloud computing technology advanced as they provide services of reserving some space in their servers to perform business core application regardless of its architecture. They have introduced their reliability of usage by ensuring a fault tolerant cloud computing services; vendors now have datacenters in a region and have multiple regions so businesses can trust even if a data center is affected by unfortunate events such as storms that their application would run as expected and their data is not lost.

# What are the available cloud services

There are many cloud computing platforms nowadays and the number keeps on grow. How do we know which one to choose and what are the key features of those platforms?

The following table demonstrate what are the top 5 cloud-computing platforms and compares them in terms of key features and data center zones:

|  |  |  |
| --- | --- | --- |
| Vendor | Datacenters | Key features |
| AWS | US, EU, MIDDLE EAST | * Easy sign-up process * Fast Deployments * Allows easy management of add or remove capacity * Access to effectively limitless capacity * Centralized Billing and management * It is one of the cloud companies that offers Hybrid Capabilities and per hour billing * Support Serverless framework intuitively * Flexible database including both NoSQL and SQL database that support scaling on demand * Secure authentication and authorization services that can integrate with 3rd party identity providers such as google, Facebook. * Provide Juptyer notebooks with accessibility of increasing machine RAM & Disk space if needed. * Over 200+ services can be integrated with serverless computing. * Various messaging brokers services. * Intuitive domain control with easy steps to get started. * Great storage choices, such is simple storage service (S3) and can be used to deploy frontend with HTTP protocol * Advanced services support HTTPS protocol for security measurements via Cloudfront. |
| Digital Ocean | USA, Canada, Australia, Germany, India, UK, Netherlands | * *Allows you to deploy your custom image, one-click app, or standard distribution* * *Option to select Standard Plans or Performance Plans according to your business needs* * *Can deploy directly from GitHub* * *Can auto-deploy once Github repository is updated.* * Support kubernates. * 200 Dollars’ worth of digitalOcean credit * Customer surveys with participation rewards |
| Azure | USA, Middle east, India, Korea, China, Australia, Brazil | * Offers the most effective solution for your data needs * Provides scalability, flexibility, and cost-effectiveness * Offers consistency across clouds with familiar tools and resources * Allow you to scale your IT resources up and down according to your business needs * Support serverless framework but require additional configurations in comparison with AWS * Provides SQL databases |
| Linode | USA, German, Mumabi, Singapore | * *Affordable Shared plans for general workloads include generous amounts of free bundled transfer.* * *Dedicated and High Memory instances on their own CPU cores.* * *On-demand GPUs for machine learning, scientific computing, and video processing.* * *24/7/365 human support.* * *S3-Compatible Object Storage and scalable Block Storage.* * *$100 in free credits during a 60-day trial.* * *Discounts up to 15%* |
| GCP | US, ASIA, EUROPE | * *It is one of the cloud companies that allows you to scale with open, flexible technology* * *Solve issues with accessible AI and Data analytics* * *Eliminate the need for installing costly servers* * *Allows you to transform your business with a full suite of cloud-based services* * Start with 300 Dollars in GCP credit per account. * API accessibility to services such as Google Map API * Support Kubernates * Support shell commander per account for additional configurations * Support serverless but require additional configurations |

# Which cloud service is suitable for OFS?

Before I worked on this research, I had some prior experience with amazon cloud using from the internship I took in semester 5 in a couple of service such as **SageMaker** for artificial intelligence modelling and **simple storage service** for automating PDF data extraction using python.

After the advice I received from Chief Technology officer of Insify I realized that serverless is a must learn for me as it allows businesses to focus on core business logic implementation and completely leave managing servers to the cloud and it supports microservices, event-driven architectural designs by nature.

Most of **serverless framework** resources online are outdated and learning how to get started with serverless is unclear for beginners, it heavily relies on experimenting and reading documentation carefully. Serverless framework support AWS vendor by default and can be integrated with **SAM** (Serverless-application-model) AWS documentation configuration, in addition to this, the group project I work on had been hosted on AWS which encouraged.

For those reasons I believe that utilizing AWS for the core backend serverless application would be ideal for me. In addition, I do need the Google map API service at some point, so I can **terraform** in the future to implement some features in AWS and GCP.

# Function as a service

Serverless frameworks work by creating what so called Functions as a service or FaaS. It implies writing some business logic in a function, we could think of it as a snippet of code that gets triggered by events. The events can be in many different forms including the most prevalent form of trigger HTTP Request, or message broker topic updates, or on a schedule so for example, every day implement some snippet of code that does some security checks daily.

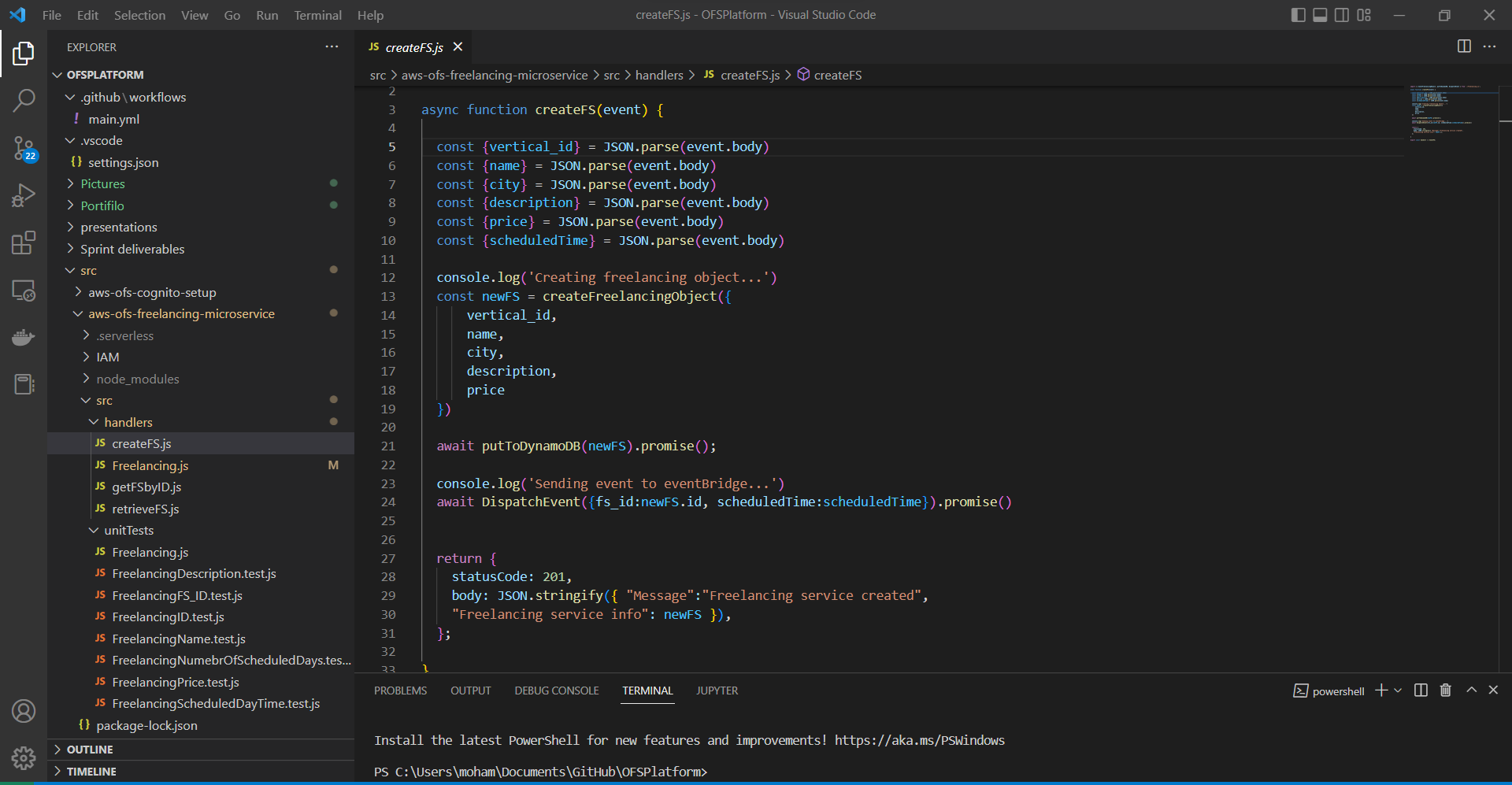
This service allows the invocation of snippets of code on the fly and can scale up on demand (AWS Lambda). Therefore, using FaaS support horizontal scaling by default.

Typically, a completely solution would use a bunch of those functions and AWS offers first million requests to be free and then every million request to be 1 dollar which is super great for startups.

Utilizing such service would help scale out and in on the fly with cost-efficient (I personally have consumed about 500k requests in 3 months so my total usage so far is 0 euros for Lambda!)

FaaS snippet of codes can be analyzed through static code analysis to determine whether dependencies are vulnerable or not using tools such as Snyk, so it supports CI/CD development cycle process. Good to note that FaaS are stateless so when they finish running some FaaS snippet of code they usually lose their state.

Below is one example of OFS Functions as a service:



# Infrastructure as a service

Infrastructure as a service follows some concept of FaaS by writing snippets of code but instead of invoking those snippets over and over by triggers, we use those IaaS (infrastructure as a service) code snippets to configure how a vendor services configuration should be.

For example, AWS offer more than 200 services, and their website interface allows for managing resources and controlling services. However, what happens if you have one root account with one IAM account for each developer (so basically all developers have access to the same amazon cloud environment) if they were to change how a table look like in the same environment there will be conflicts and there is room for human error. This is why controlling how the environment should look like via code is the modern solution and we need to keep in mind

# Services cost analysis

# Conclusion