CMC GLOBAL

Aspire to Inspire the Digital World

Cloud Professional Service
CMC Global - 2020

Case studies

CASE STUDY 1: Cloud Migration Service

Massive Cloud Migration

Industry

Japanese corporation
Founded in 1951. The company's line of business include operating nonresidential buildings. 2645 employees as Mar 2019

Challenges

Client is currently host their infrastructure system on premise. This causes them several troubles such as: high cost of ownership compared to Cloud-based system, hard to scale, limited access area & Limited on disaster recovery ability,… Therefore, Client request CMC Global to move all of their enormous system – which contains nearly 1000 VMs and other services - to Cloud environment

CASE STUDY 1: Cloud Migration Service

Our Solution and benefits

To effectively migrate the immense system of Client, CMC Global implement a five-phases migration process:

1

Migration preparation and business planning

After consider the pros and cons of Cloud computing services and the characteristics of Client’s system, we consult our customer to choose AWS

2

Portfolio discovery and planning

CMC Global apply Decision framework to consult our customer on migration process

3

Designing, provisioning configuration management

After consider the pros and cons of Cloud computing services and the characteristics of Client’s system, we consult our customer to choose AWS

4

Migrating, validating infrastructure and applications

5

Modern operating mode

Approximately 1000 VMs and services of client corporation was migrated to AWS

Client’s cost of ownership and operation cost reduced 30% in compared with old on - premises system.

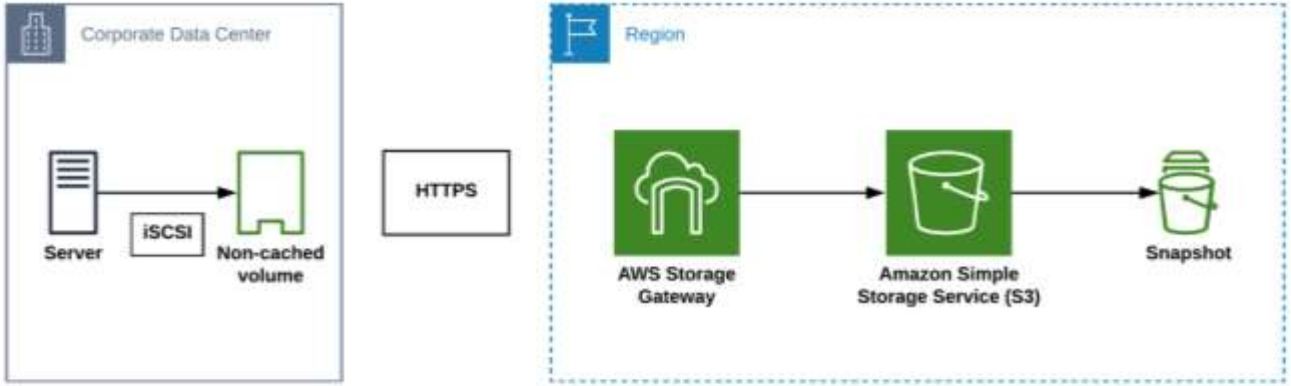
CASE STUDY 2: Cloud Migration Service

Provisioning secure remote file access on AWS with Azure

Industry

Electrical, Telecommunication

The company maintains a large distributed workforce. Many of these employees require access to the company's business analytics service and documents but are not connected to the company's secure, on premise network

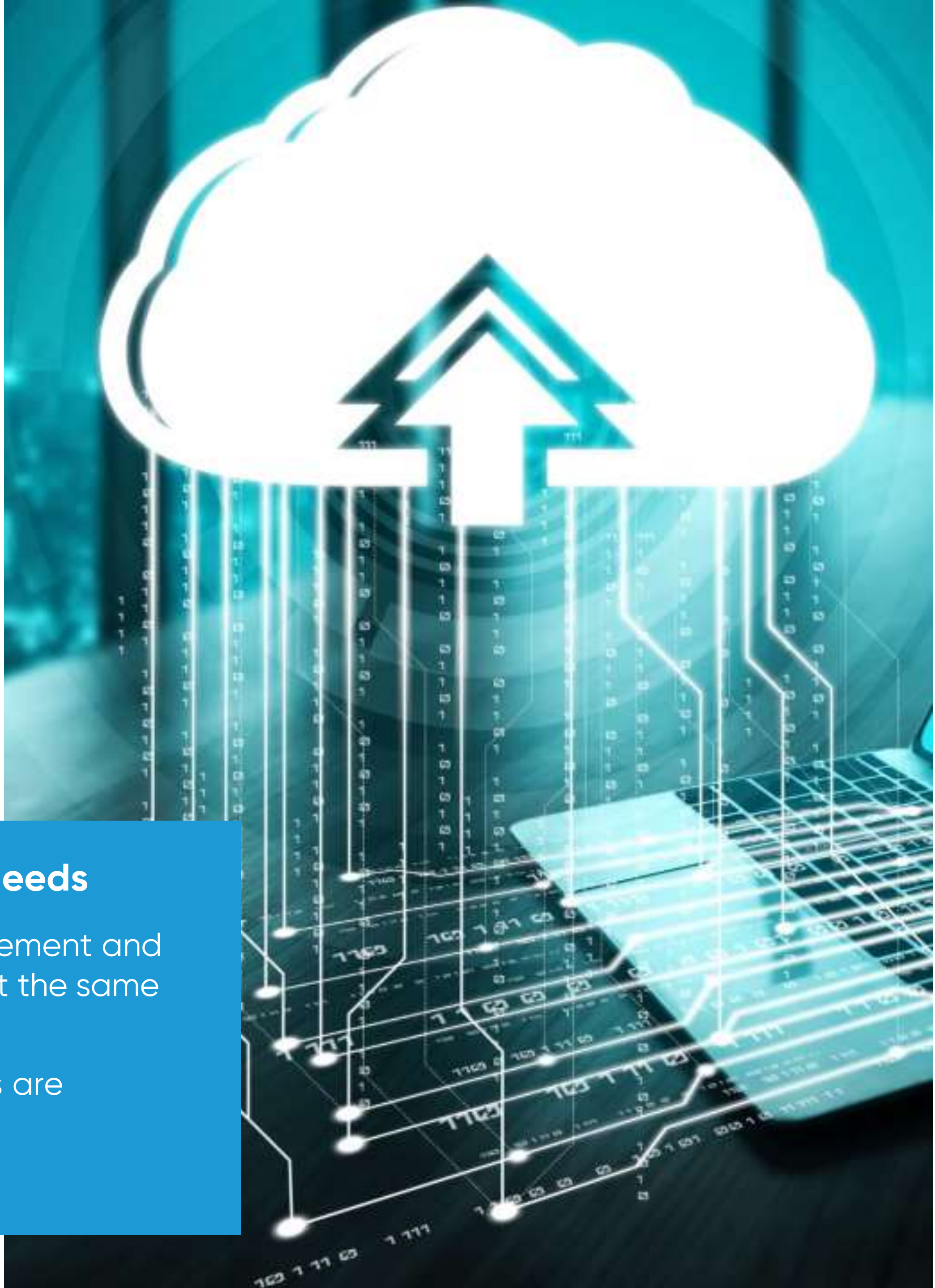


CASE STUDY 2: Cloud Migration Service

Challenges

The company employs a large staff of employees on-site and at project locations that often need to access PDF documents from the company's Microsoft Power BI online business analytics service.

At a remote location, employees are not on the company's network. That makes accessing those documents a potential security risk for company and a tedious, time-consuming process for the employees.



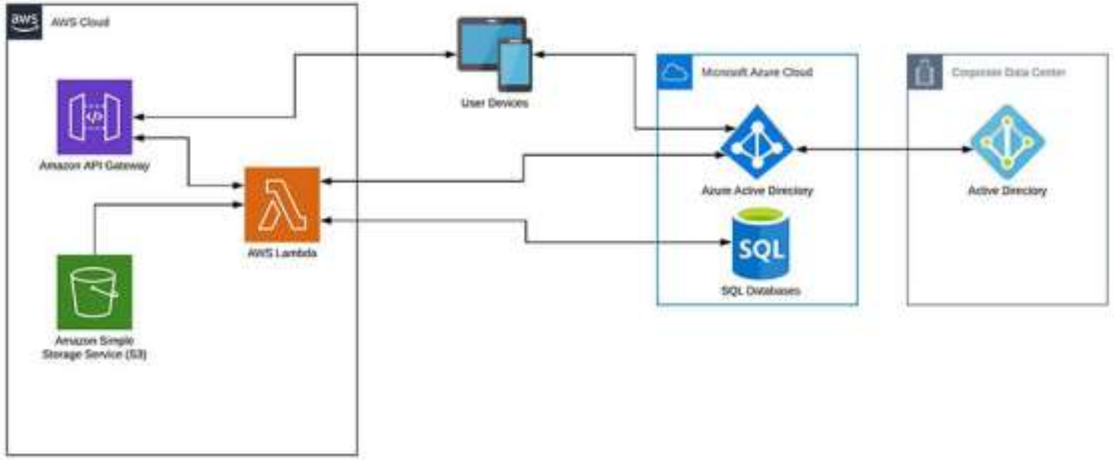
Cloud Storage solution needs

- Simplifies storage management and reduce cost for storage at the same time
- Ensure security constrains are strictly followed

CASE STUDY 2: Cloud Migration Service

Our Solution

- Users are authenticated with Azure-AD
- User access to PDF file in S3 bucket with lambda function
- URL to access the files are valid for subscribed users and a short period of time only



Benefits

By integrating AWS storage services with Office365 and Azure AD, our solution ensures access to documents in Power BI is restricted to only those employees with authorization.
Access the needed resources quickly without additional user authentication steps and remain secure.
Increase file load performance by using lambda function.

CASE STUDY 3: Cloud Migration Service

AWS server-less architecture design

Industry

E-commerce, over 10000 active users

Challenges

Design online shopping site to sell products from a number of categories, including consumer electronics, household goods, toys, fashion, sports equipment and groceries.

Apart from the current hybrid platform running on both cloud and on premise, client would like to build a whole new platform on AWS can adapt rapid changes in order to tailor the website base on multiple regions culture

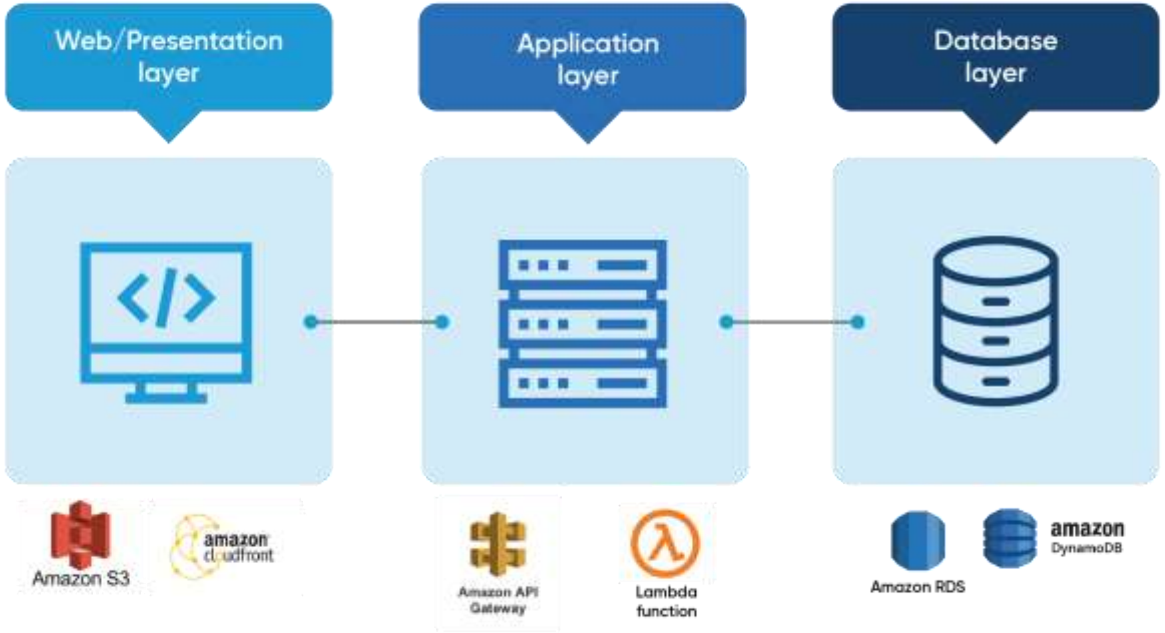
CASE STUDY 3: Cloud Migration Service

Our solution

Architecture design

S3 bucket for hosting static web content

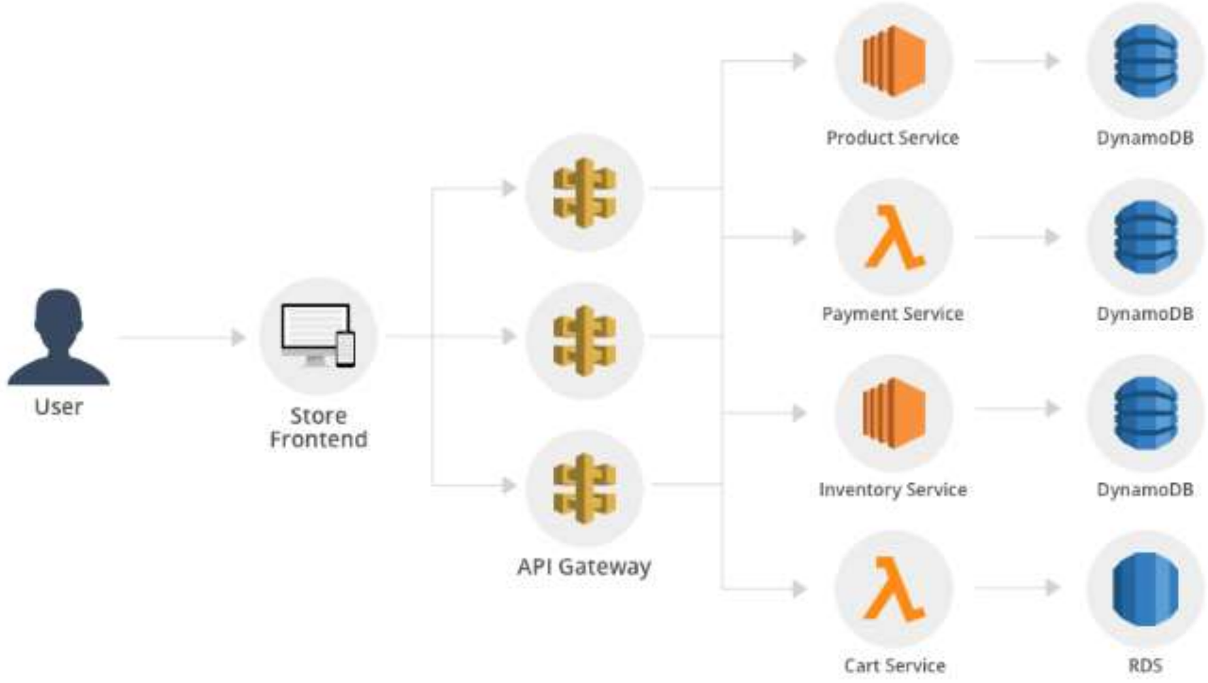
- AWS CloudFront for content caching, provide lower latency, faster response
- Provide HTTPS which is not supported by S3
- Authenticate user by using Lambda@Edge
- API Gateway as front end for different services exposed by the application
- AWS Lambda as compute engine for application backend which provide elasticity and seamlessly integrated with API gateway
- Amazon RDS: Highly Available scalable DBaaS support multiple DB engines
- Provide private endpoints for data connections
- Using Dynamo DB for large data processing



CASE STUDY 3: Cloud Migration Service

Benefits

- Enable current platform to call API securely
- Operational cost is kept at minimum to be compared with on-premise platform
- Security in every component of the architecture, end-to-end encryption starting from client side.



High level architecture design

CASE STUDY 4: Development and DevOps Service

Client Overview

- **Industry**: Healthcare site, provide healthcare reservation service for a clinic agent with hundred of employee.
- Our client runs its entire cloud-based IT infrastructure on AWS

Needs and Challenge

- Robustness of service with high levels of uptime and reliability
- Focus on open source tools like Jenkins Packer and Docker
- Completely automated so developers could focus on software rather than managing servers

CASE STUDY 4: Development and DevOps Service

Challenges & solutions

1 Using containers and a microservice architecture together with automation enhanced their cloud capabilities. Microservices are scalable and reusable, while containers supply efficient resource management. Both microservices and containers can work independently, but it has become clear that merging them has improved runtime frequency, deployments and overall application efficiency.

2 Running microservices-based applications on Amazon Elastic Kubernetes Service (Amazon EKS) are looking for guidance on architecting complete end-to-end Continuous Integration (CI) and Continuous Deployment / Delivery (CD) pipelines using Jenkins

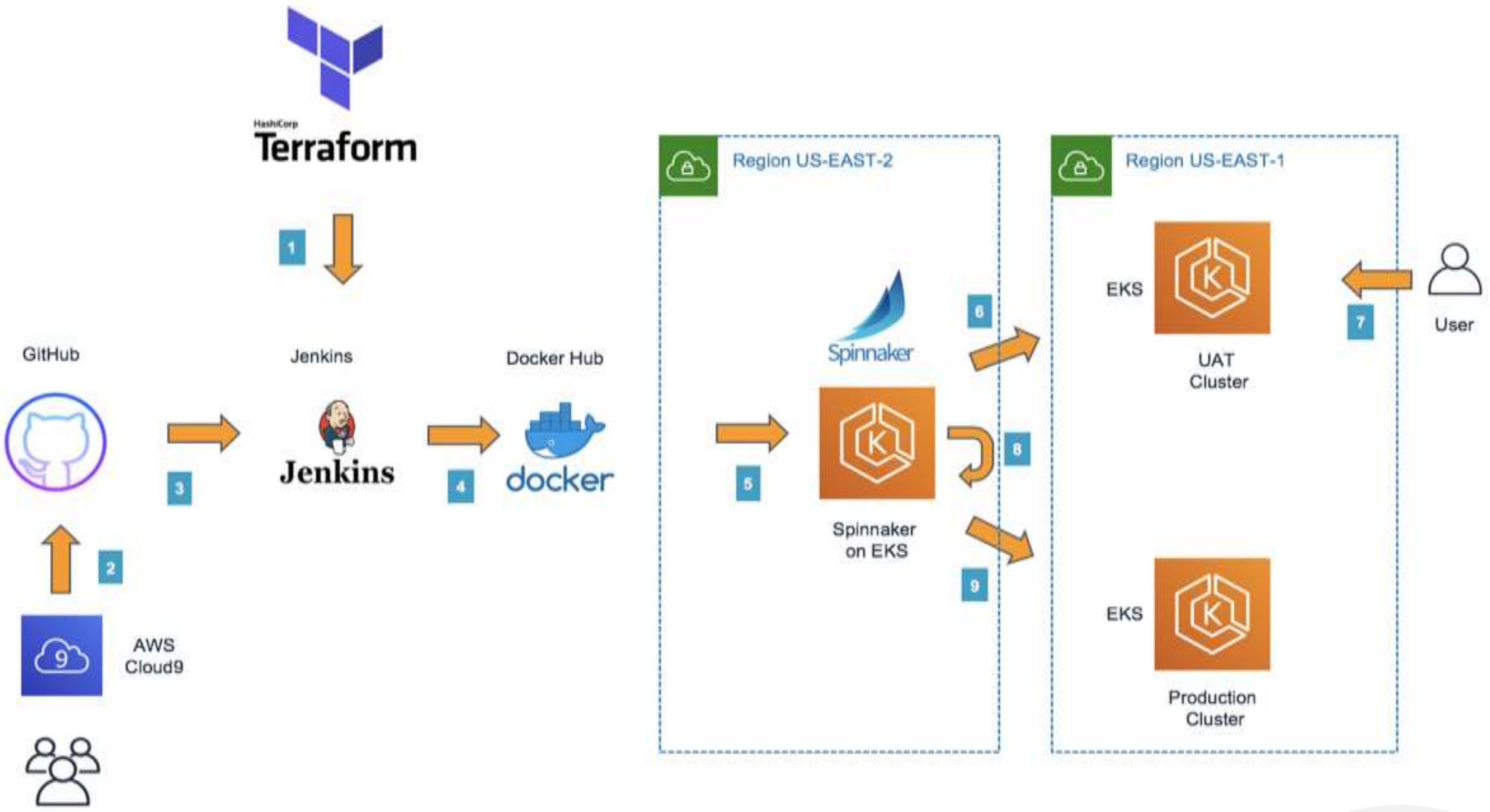
3 The time taken from development phase to project go-live was reduced to 2 months. Giving rise to business agility, Customer can now react faster to changes in the market.

4 Deploying applications to the cloud meant our Customer's application was running closer to their customers, resulting in lower latencies and a better overall user experience.

5 Our customers have the assurance of a higher service availability because the environment leverages autoscaling components in AWS. By using additional AWS services, They can also release new code reliably and more effectively, without impacting the service.

CASE STUDY 4: Development and DevOps Service

AWS continues integration model



1 Provision Jenkins server using Terraform

2 Developers integrate code changes into a code repository

3 Code changes trigger Jenkins Pipeline

4 Jenkins build a new docker image and pushes it with a build id to Docker registry

5 Docker image triggers the first Spinnaker pipeline

6 Spinnaker performs a deployment into UAT cluster

7 UAT testing

8 Spinnaker Human/Manual approval pipeline

9 Spinnaker performs a deployment into Production cluster

CASE STUDY 5: Development and DevOps Service

Migrating from monolithic to micro services

Industry

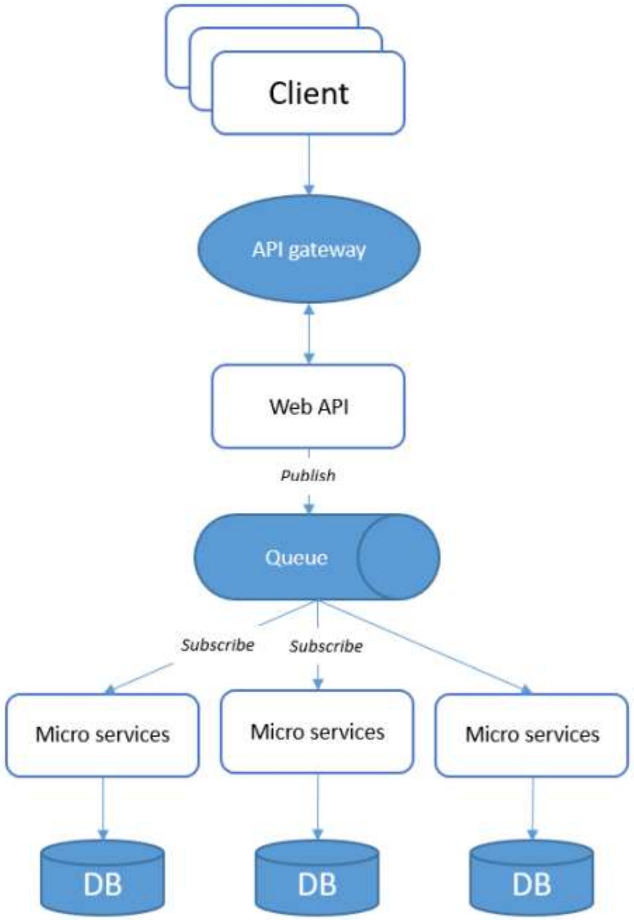
Electrical
The client is one of largest electronic component distributor in the world.
Our client has built an ERP System to serve retail that is written in Java with over millions of lines of code. Their system has been running for almost 20 years

Challenges

- Many technologies have been developed more than 20 years ago have become outdated.
- The original design was a monolithic design that became cumbersome due to too many functions added, difficult to maintain and the process of building and implementing source code takes a lot of time.
- Deploying to expand the system is really a head-ache task.
- Database of our customer is very complicated, it is the challenge for us to divide into multi-database in the microservices architecture
- They require us the continuance of the system during the migration process

CASE STUDY 5: Development and DevOps Service

Migrating from monolithic to microservices



The business of our customer is Distributed management system. Therefore, the domain drive design method (DDD) was applied to analyze and divide the system into separated modules such as: Order module, Sale module, etc.

From the result of DDD analysis, we counted the number of modules and estimate the microservice needed.

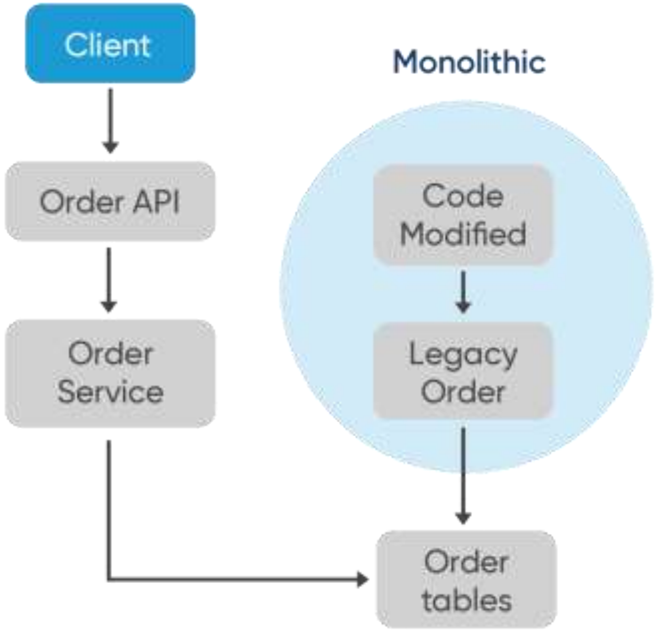
The microservices is separated into internal service and external service. We use AWS ECS to run application containers. The internal service directly interacts with the database, sending and receiving asynchronous data to the external service (expose API) to the client via a message queue.

The API section uses the Swagger to generate API specs. We used the method of dividing and converting data into parts such as collecting tables in sale, one database owned by the sale services, the payment into one database, owned by the payment service.

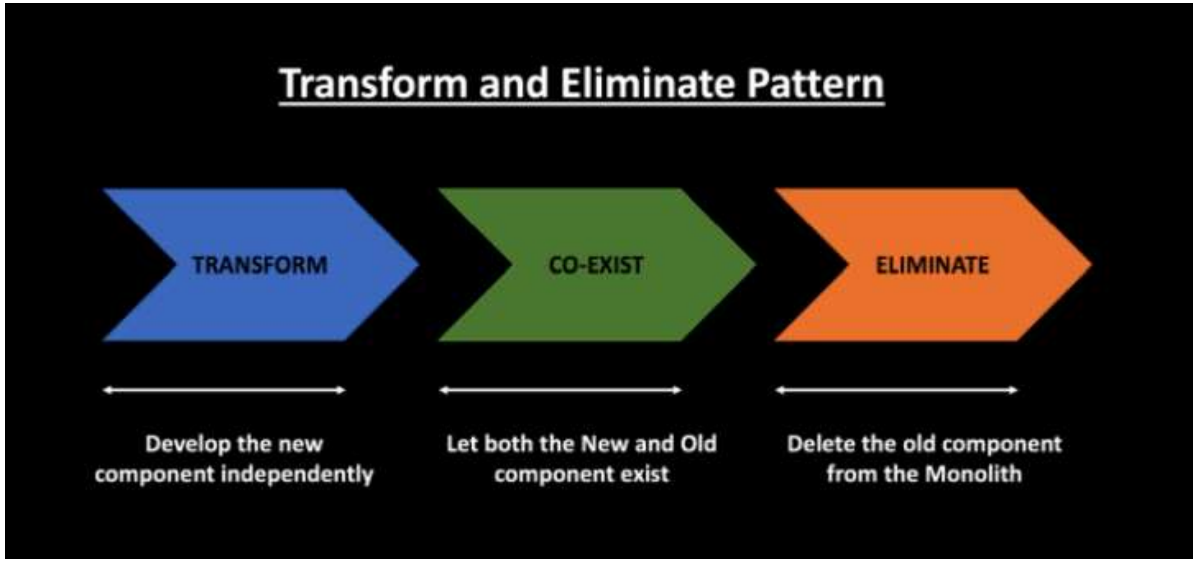
From the old business analysis system, we peel out a specific business function (e.g.: order, payment). We create Bounded Context - a border around that business logic and data, the other modules want to access, they must go through the interface.

CASE STUDY 5: Development and DevOps Service

Migrating from monolithic to micro services



For example, the Order Service is separated into a standalone service from the original monolithic. Determining bounded context Order from monolithic separated into Order microservice, the code calling legacy order via function call is modified to switch to call by API (order API).

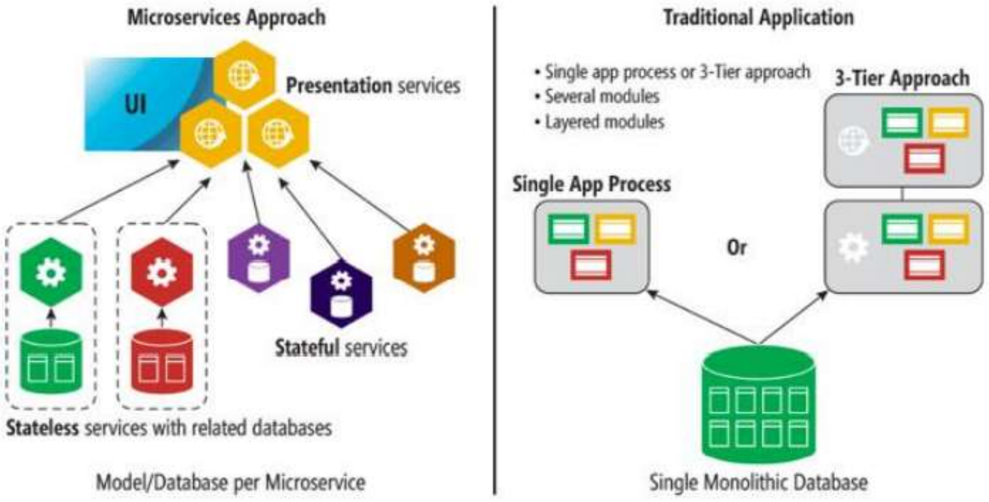

Transform and Eliminate Pattern
TRANSFORM
CO-EXIST
ELIMINATE
Develop the new component independently
Let both the New and Old component exist
Delete the old component from the Monolith

Once the Bounded Context is defined, the next step is to create the interface. In most cases the interface is designed to be Rest API. External modules will be adjusted to reach bounded context via interface.

Therefore, every time the module is split into a microservice, the rest of the mono system will be modified to call them through the interface instead of the function call as before. The process of splitting and repairing is done slowly, step by step, until the entire migration system is complete. The the old and new systems co-exist until the migration is completed.

CASE STUDY 6: Development and DevOps Service

Migrating from monolithic to micro services



The final step is how to break down the monolithic DB into its own DB by own service. The challenge is how to migrate data from mono DB to microservices DB. Because data is an organization's property, not an application's, data reaches millions of records with hundreds of interrelated tables.

Based on the analysis that we have done so far, we have solved this challenge with the help of data synchronization pattern and use database migration service of AWS to migrate the database.

After migrate to microservices, our customer gains the flexibility benefit: the system now is not only written by Java, but also services that are written in Python, services to store cache only, have route gateway service, etc.

CASE STUDY 6: Application Development Service

Migrating a legacy app to a serverless ecosystem

Industry

A Japanese IT service company who provides business solutions using IT technology. Headquarter: Koto, Tokyo, Japan. They are a group company of Mitsui group.

Challenge

Working on a project that Migrate from VB6 in Oracle application, modernizing it in the cloud, but instead of doing the lift and shift move it to an EC2, they decided to simply jump to service.

We migrate from VB6 in Oracle application, modernizing it in the cloud and from RDS to DynamoDB. When working, the old app is still being used on a day to day basis & we're progressively adding the feature & re-implementing in the cloud. Therefore:

- Need to synchronize our data & for that using DMS and a few different things to push to our DB
- We publish that over SNS & Lambda a like service will pick that up and finally format it for DynamoDB

CASE STUDY 7: Application Development Service

Challenges & solutions

1 We first went into project, relational DB, now is DynamoDB => have to think in different ways and make a few hops & identified there's a lot of data that could be relational wise & the way to do that in DynamoDB is in adjacency list patterns, that many - to - many data one of many data & combine it together into one table.

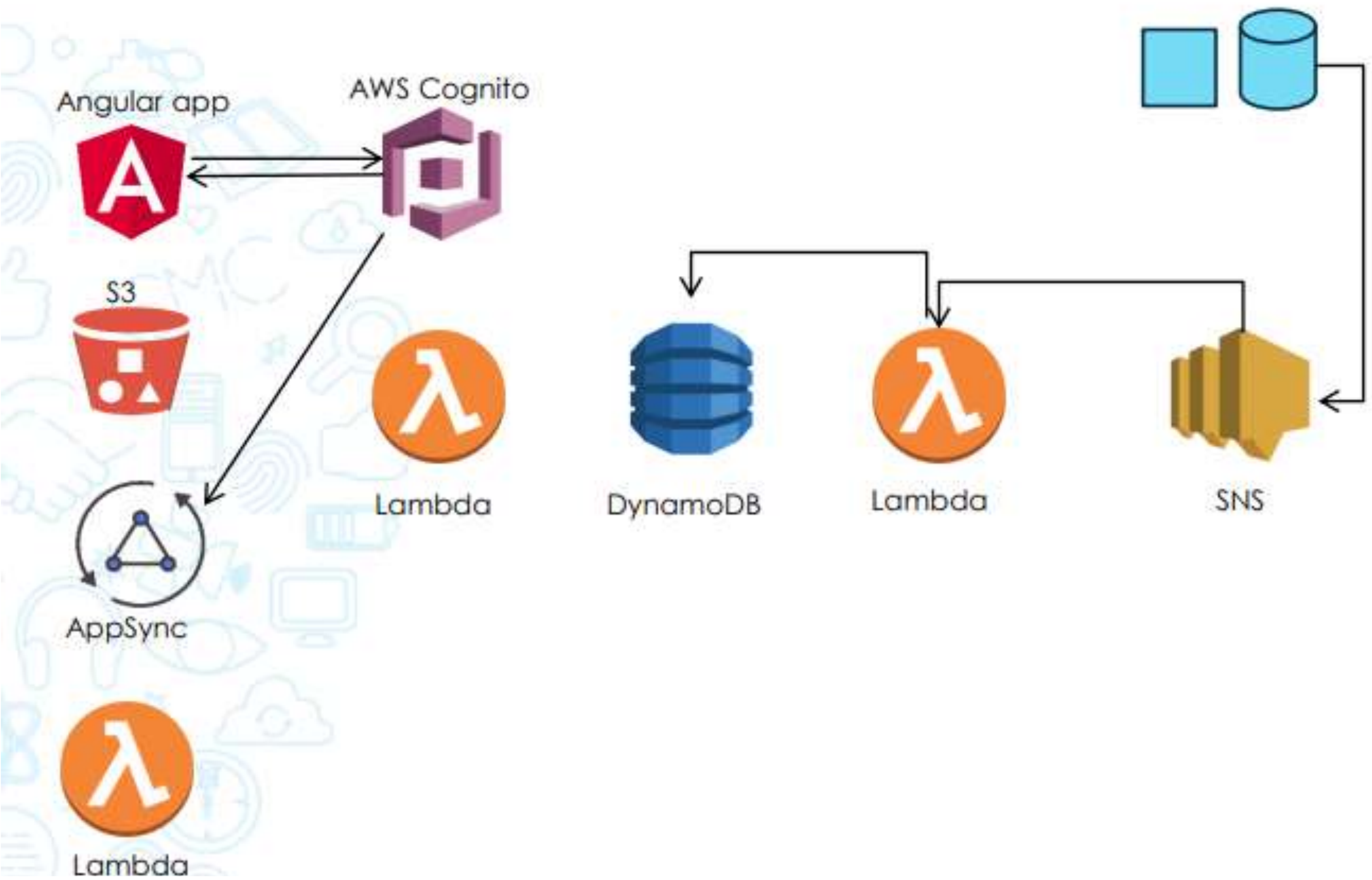
2 Find a partition key for something like a vendor may contain the data itself about vendor info but may have other things like operation area, address,.. => all this kind of data is combined together under 1 specific partition key.

3 Data points coming out lambda bottle & that this micro-services is coming up into AppSync & it's been married by legacy data

4 AppSync: good for combining multiple kinds of different data together it provide interface and allow flexibility to grow where we have queries of mutations that reaching out to one micro-services start adding a new schemas from another different microservices & it start to marry those together as a single interface.

5 AWS Cognito: Control authorization for the app Benefit: Allow end users to query DB without understanding sequel or data structure & they can just understand the nodes & helping interrelate to one another.

CASE STUDY 7: Application Development Service



THANK YOU

Aspire to Inspire
the Digital World

HEAD OFFICE

8th-9th Floor, CMC Tower, Duy Tan Str.
Dich Vong Hau Ward, Cau Giay Dist., Hanoi
www.cmcglobal.com.vn

DANANG

8F, Thanh Loi II Tower, 1 Le Dinh Ly
Street, Vinh Trung Ward, Thanh Khe
District, Da Nang City

HCM CITY

3F, ITD Building, 1 Sang Tao Street,
Tan Thuan EPZ, Tan Thuan Dong Ward,
District 7, Ho Chi Minh City

CMC JAPAN

Ocean Gate Building 8F, 3 Chome–7–1,
Minatomirai, Nishi Ward,
Yokohama 220-0012, Japan

+84 24 7109 6686
+84 24 3212 3396
cmcglobalinfo@cmc.com.vn