

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**LÊ QUANG HOAN**

**KIẾN TRÚC PHẦN MỀM CHỊU TẢI CAO DỰA TRÊN  
NỀN TẢNG ĐIỆN TOÁN ĐÁM MÂY MICROSOFT  
AZURE**

**LUẬN VĂN THẠC SĨ KỸ THUẬT PHẦN MỀM**

**HÀ NỘI – 2018**

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**LÊ QUANG HOAN**

**KIẾN TRÚC PHẦN MỀM CHỊU TẢI CAO DỰA TRÊN  
NỀN TẢNG ĐIỆN TOÁN ĐÁM MÂY MICROSOFT  
AZURE**

**LUẬN VĂN THẠC SĨ KỸ THUẬT PHẦN MỀM**

**Cán bộ hướng dẫn: TS. Trần Trọng Hiếu**

**Cán bộ đồng hướng dẫn: PGS. TS. Phạm Ngọc Hùng**

**HÀ NỘI - 2018**

## LỜI CẢM ƠN

Đầu tiên, tôi xin gửi lời cảm ơn chân thành tới TS. Trần Trọng Hiếu – giảng viên Bộ môn Các Hệ thống thông tin và PGS.TS. Phạm Ngọc Hùng - giảng viên Bộ môn Công nghệ phần mềm – những người đã tận tình hướng dẫn tôi trong suốt thời gian làm luận văn tốt nghiệp. Quãng thời gian được các thầy hướng dẫn đã giúp tôi học hỏi, đúc kết được nhiều kinh nghiệm về phương pháp nghiên cứu, kỹ năng giao tiếp, kỹ năng làm việc nhóm, kỹ năng trình bày.

Tôi xin gửi lời cảm ơn đến các thầy cô Khoa Công nghệ thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội nói chung cũng như các thầy cô trong bộ môn Công nghệ phần mềm nói riêng đã tận tình giảng dạy tôi trong suốt thời gian tôi tham gia học tập tại trường.

Cuối cùng, tôi xin được gửi lời cảm ơn chân thành tới gia đình, bạn bè cũng như các đồng nghiệp đã luôn ủng hộ, động viên tôi để tôi có thể có điều kiện học tập và nghiên cứu.

Hà Nội, ngày    tháng    năm 2018

Học viên

Lê Quang Hoan

## **LỜI CAM ĐOAN**

Tôi xin cam đoan rằng những nghiên cứu về kiến trúc phần mềm triển khai trên nền tảng điện toán đám mây Microsoft Azure được trình bày trong luận văn này là của tôi và chưa từng được nộp như một khóa luận, luận văn hay luận án tại trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội hoặc bất kỳ trường đại học khác. Những gì tôi viết ra không sao chép từ các tài liệu, không sử dụng các kết quả của người khác mà không trích dẫn cụ thể. Nếu sai tôi hoàn toàn chịu trách nhiệm theo quy định của trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội.

Hà Nội, ngày    tháng    năm 2018

Học viên

Lê Quang Hoan

## MỤC LỤC

### Giới thiệu 1

<b>Chương 1. TỔNG QUAN VỀ ĐIỆN TOÁN Đám Mây</b>	<b>3</b>
1.1. Điện toán đám mây	3
1.1.1. Khái niệm	3
1.1.2. Các đặc tính cơ bản của điện toán đám mây	4
1.2. Các mô hình dịch vụ trong điện toán đám mây	5
1.2.1. Dịch vụ hạ tầng (IaaS – Infrastructure as a Service)	5
1.2.2. Dịch vụ nền tảng (PaaS – Platform as a Service)	7
1.2.3. Dịch vụ phần mềm (SaaS – Software as a Service)	8
1.3. Các thành phần của điện toán đám mây	9
1.4. Các mô hình triển khai điện toán đám mây	10
1.4.1. Mô hình đám mây riêng (Private Cloud)	10
1.4.2. Mô hình đám mây công (Public Cloud)	11
1.4.3. Mô hình đám mây lai (Hybrid Cloud)	12
1.4.4. Mô hình đám mây cộng đồng (Community Cloud)	13
1.5. Kết luận	13
<b>Chương 2. KIẾN TRÚC PHẦN MỀM DỰA TRÊN CÁC DỊCH VỤ ĐIỆN TOÁN Đám Mây MICROSOFT AZURE</b>	<b>15</b>
2.1. Nền tảng Microsoft Azure	15
2.1.1. Tổng quan về Window Azure Platform	15
2.1.2. Nền tảng Microsoft Azure	16
2.2. Các kiểu kiến trúc phần mềm trên Cloud	23
2.2.1. Kiến trúc phân tầng (N-tier)	23
2.2.2. Kiến trúc Web - Queue - Worker	26
2.2.3. Kiến trúc vi dịch vụ (Microservice)	28
2.3. Các yếu tố ảnh hưởng đến khả năng chịu tải của hệ thống	31
2.3.1. Đảm bảo hiệu năng (Performance)	32

2.3.2.	Đảm bảo tính sẵn sàng của hệ thống (Availability).....	34
2.3.3.	Đảm bảo tính mở rộng hệ thống (Scalability) .....	34
2.4.	Kết luận.....	36
<b>Chương 3. MỘT MÔ HÌNH ỨNG DỤNG KIẾN TRÚC PHẦN MỀM TRÊN NỀN TẢNG CÔNG NGHỆ AZURE CỦA MICROSOFT .....</b>		<b>37</b>
3.1.	Mô tả bài toán .....	37
3.1.1.	Giới thiệu .....	37
3.1.2.	Giải pháp.....	37
3.2.	Phân tích nghiệp vụ .....	37
3.2.1.	Mô tả chức năng .....	37
3.2.2.	Quy trình chấm công bằng khuôn mặt.....	38
3.2.3.	Biểu đồ các trường hợp sử dụng (Use Case) .....	40
3.2.4.	Các module chức năng hệ thống.....	44
3.3.	Thiết kế hệ thống .....	45
3.3.1.	Mô hình tổng thể chức năng hệ thống .....	45
3.3.2.	Mô hình phân rã chức năng .....	45
3.3.3.	Kiến trúc hệ thống.....	46
3.3.4.	Quy trình xử lý dữ liệu ảnh khi nhận diện.....	49
3.4.	Xây dựng chương trình thử nghiệm .....	50
3.4.1.	Môi trường cài đặt, triển khai .....	50
3.4.2.	Các bước triển khai ứng dụng.....	51
3.4.3.	Màn hình giao diện .....	55
3.5.	Đánh giá khả năng chịu tải của hệ thống.....	58
3.5.1.	Đánh giá với số lượng user đồng thời tăng dần.....	59
3.5.2.	Kiểm thử với số lượng instance tăng dần .....	60
3.6.	Kết luận.....	62
<b>Chương 4. KẾT LUẬN.....</b>		<b>63</b>

## DANH SÁCH KÍ HIỆU, CHỮ VIẾT TẮT

Kí hiệu	Thuật ngữ	Ý nghĩa
IaaS	Infrastructure as a Service	Dịch vụ hạ tầng
Paas	Platform as a Service	Dịch vụ nền tảng
SaaS	Software as a Service	Dịch vụ phần mềm
CDN	Content Delivery Network	Mạng phân phối nội dung
ĐTĐM	Điện toán đám mây	Điện toán đám mây
CNTT	Công nghệ thông tin	Công nghệ thông tin
CRM	Customer Relationship Management	Quản lý quan hệ khách hàng
VNet	Virtual Network	Mạng ảo
SDK	Software Development Kit	Bộ phát triển phần mềm
NSG	Network Security Group	Nhóm bảo mật mạng
DMZ	Demilitarized zone	Vùng hạ tầng dễ bị tổn thương

## DANH SÁCH HÌNH VẼ

Hình 1.1: Mô hình điện toán đám mây.....	4
Hình 1.2: Các loại dịch vụ của ĐTĐM.....	5
Hình 1.3: Mô hình IaaS .....	6
Hình 1.4: Mô hình PaaS .....	7
Hình 1.5: Mô hình SaaS .....	8
Hình 1.6: Thành phần của điện toán đám mây.....	9
Hình 1.7: Mô hình đám mây riêng (Private Cloud) .....	11
Hình 1.8: Mô hình đám mây công (Public Cloud).....	11
Hình 1.9: Mô hình kết hợp Hybrid Cloud .....	12
Hình 1.10: Mô hình đám mây cộng đồng.....	13
Hình 2.1: Tổng quan mô hình Azure Platform.....	15
Hình 2.2: Các thành phần của Microsoft Azure.....	16
Hình 2.3: Mô hình dịch vụ Compute của Microsoft Azure.....	17
Hình 2.4: Dữ liệu được quản lý trong Microsoft Azure.....	18
Hình 2.5: Mô hình Networking trên Azure. ....	19
Hình 2.6: Mô hình Application Insight trên Azure. ....	19
Hình 2.7: Mô hình Active Directory trên Azure. ....	20
Hình 2.8: Mô hình sử dụng IoT Hub trên Azure.....	21
Hình 2.9: Mô hình của Media Service.....	22
Hình 2.10: Mô hình của Media Service.....	22
Hình 2.11: Mô hình Kiến trúc phân tầng [11] .....	23
Hình 2.12: Mô hình Kiến trúc phân tầng trên Azure [11].....	24
Hình 2.13: Mô hình Kiến trúc Web – Queue – Worker [11] .....	26
Hình 2.14: Mô hình Kiến trúc Web – Queue - Worker trên Azure [11].....	26
Hình 2.15: Mô hình Kiến trúc Microservice [11] .....	28
Hình 2.16: Mô hình Kiến trúc Microservice sử dụng Azure Container Service [11] ...	29
Hình 2.17: Cân bằng tải với Load Balancer .....	32
Hình 2.18: Cách thức hoạt động của CDN.....	33
Hình 2.19: Cách thức hoạt động của Caching .....	33
Hình 2.20: Kiến trúc Master/ Slave trong Azure SQL .....	34
Hình 3.1: Quy trình chấm công bằng khuôn mặt .....	38
Hình 3.2: Các trường hợp sử dụng tổng quan .....	40
Hình 3.3: Use case Quản lý phòng ban .....	41



Hình 3.4: Use case Quản lý nhân viên .....	41
Hình 3.5: Use case Quản lý lịch làm việc .....	42
Hình 3.6: Use case Chăm công.....	42
Hình 3.7: Use case Quản lý người dùng.....	43
Hình 3.8: Use case Cấu hình hệ thống .....	43
Hình 3.9: Use case Thống kê.....	44
Hình 3.10: Mô hình tổng thể chức năng hệ thống.....	45
Hình 3.11: Mô hình phân rã chức năng.....	45
Hình 3.12: Kiến trúc hệ thống .....	46
Hình 3.13: Quy trình xử lý ảnh khi nhận diện.....	49
Hình 3.14: Hình ảnh bố trí Camera .....	54
Hình 3.15: Ảnh mẫu nhận diện nhân viên.....	55
Hình 3.16: Giao diện theo dõi người vào ra .....	55
Hình 3.17: Giao diện đăng nhập phần mềm.....	56
Hình 3.18: Giao diện thông tin chăm công .....	56
Hình 3.19: Giao diện thống kê lượt vào ra .....	57
Hình 3.20: Giao diện tổng hợp, thống kê .....	57
Hình 3.21: Giao diện sử dụng Application Insight .....	58
Hình 3.22: Giao diện kết quả khi chạy load test .....	58
Hình 3.23: Biểu đồ số thể hiện thời gian thực hiện yêu cầu – test 1 .....	59
Hình 3.24: Biểu đồ số request thực hiện được trong 1 giây – test 1 .....	60
Hình 3.25: Biểu đồ số thể hiện thời gian thực hiện yêu cầu – test 2.....	61
Hình 3.26: Biểu đồ số request thực hiện được trong 1 giây – test 2 .....	61

## DANH SÁCH BẢNG

Bảng 2.1: Bảng thông số cấu hình của dịch vụ Azure Web app .....	35
Bảng 2.2: Bảng thông số instance tối đa có thể mở rộng của Azure Web app .....	35
Bảng 3.1: Kết quả thực hiện load test 1.....	59
Bảng 3.2: Kết quả thực hiện load test 2.....	61

## Giới thiệu

Ngày nay, việc ứng dụng CNTT ngày càng nhiều và ảnh hưởng sâu sắc đến mọi mặt của đời sống, chúng ta có thể thấy CNTT được áp dụng trong hầu hết các lĩnh vực của cuộc sống, từ trò chơi điện tử được thiết kế để đáp ứng nhu cầu giải trí, các ứng dụng soạn thảo văn bản hỗ trợ việc xử lý các công việc văn phòng, đến các ứng dụng áp dụng học máy trong các lĩnh vực tài chính ngân hàng. Chúng ta có thể dễ dàng thấy được những ưu điểm của việc ứng dụng CNTT hiện nay, đó là đưa mọi người trên khắp thế giới tiến lại gần nhau hơn.

Cùng với sự phát triển các yêu cầu người dùng, các hệ thống máy tính ngày càng trở nên lớn hơn, phức tạp hơn và phân tán hơn. Mục đích để đáp ứng được số lượng lớn người dùng, có khả năng chịu tải cao hơn. Để thỏa mãn nhu cầu này, ban đầu người ta thường phải đầu tư những hệ thống tốn kém. Làm thế nào để giảm đi những chi phí không cần thiết mà vẫn thỏa mãn được nhu cầu sử dụng là một bài toán luôn được đặt ra. Điện toán đám mây là một trong những giải pháp tốt nhất để giải quyết được bài toán trên. Với những gì mà điện toán đám mây mang lại, có thể nói rằng một cuộc cách mạng lớn trong lĩnh vực CNTT đã ra đời.

Điện toán đám mây là khái niệm hoàn chỉnh cho xu hướng nhiều doanh nghiệp hiện nay không có máy chủ riêng mà chỉ có máy tính với một số phần mềm cơ bản, còn tất cả đều phụ thuộc vào đám mây. Với các dịch vụ có sẵn trên internet, doanh nghiệp không phải mua và duy trì hàng trăm, hàng nghìn máy tính cũng như các phần mềm kèm theo mà họ chỉ cần tập trung công việc của mình bởi đã có người khác lo cơ sở hạ tầng và công nghệ thay họ.

Microsoft là một trong những nhà cung cấp dịch vụ điện toán đám mây hàng đầu thế giới hiện nay, trong đó Azure là một nền tảng chiến lược của Microsoft. Azure cung cấp cho lập trình viên nhiều tiện ích và hạ tầng để xây dựng các ứng dụng trên nền web.

Trong luận văn này, chúng tôi tập trung trình bày những khái niệm tổng quan về điện toán đám mây Microsoft Azure và tìm hiểu các dịch vụ được cung cấp của Azure. Các kiến trúc phần mềm trên nền tảng đám mây, đồng thời xây dựng một kiến trúc phần mềm, kết hợp các dịch vụ của Azure cho một ứng dụng chịu tải cao. Luận văn được trình bày trong 4 chương:

Chương 1: Giới thiệu những khái niệm cơ bản về điện toán đám mây, kiến trúc, đặc tính, thành phần của điện toán đám mây.

Chương 2: Kiến trúc phần mềm dựa trên các dịch vụ của điện toán đám mây Microsoft Azure.

Chương 3: Một mô hình ứng dụng kiến trúc phần mềm trên nền tảng công nghệ Azure của Microsoft.

Chương 4: Tóm tắt kết quả thu được qua luận văn.

# **Chương 1. TỔNG QUAN VỀ ĐIỆN TOÁN Đám Mây**

## **1.1. Điện toán đám mây**

### **1.1.1. Khái niệm**

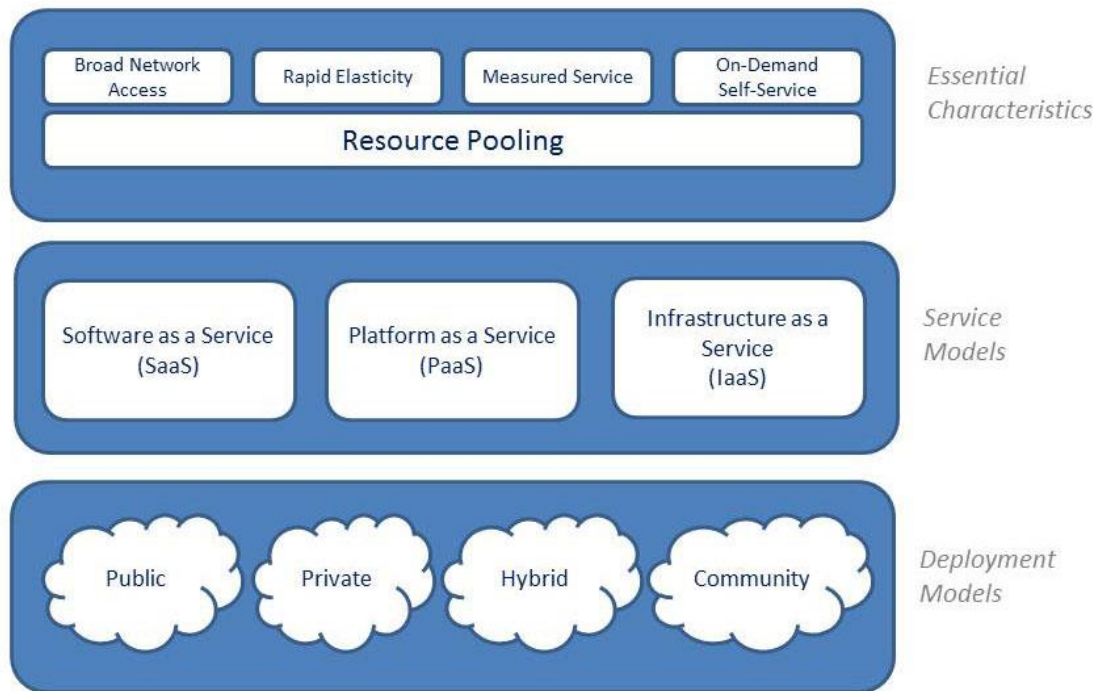
Hạ tầng máy tính, viễn thông ngày nay có thể nói là đã hội tụ trên nền tảng công nghệ số. Với các công nghệ kết nối có thể kể đến như: skết nối có dây, không dây, kết nối qua cáp đồng, cáp quang, vệ tinh, wifi hay mạng 3G, 4G,... cho phép kết nối mạng toàn cầu, vươn tới mọi nơi trên thế giới. Hạ tầng cơ sở kỹ thuật công nghệ phát triển dẫn đến các thiết bị tính toán cũng hết sức đa dạng, từ các siêu máy tính, máy chủ lớn, tới các máy tính cá nhân, máy tính xách tay, các thiết bị di động thông minh hay các thiết bị di động giá rẻ đều có thể kết nối với nhau.

Trong thế giới điện toán, khi các thiết bị đã được kết nối với nhau thì làm thế nào để khai thác được tối đa năng lực điện toán đó với chi phí thấp nhất và thời gian nhanh nhất? Các nhu cầu đặt ra là vô cùng to lớn và điện toán đám mây (Cloud computing) ra đời được kỳ vọng sẽ đáp ứng được tất cả các yêu cầu trong thực tế của con người. Điện toán đám mây sẽ giúp đem các sản phẩm và dịch vụ công nghệ thông tin chất lượng cao đến mọi đối tượng theo nhu cầu, với thời gian nhanh và chi phí rẻ hơn.

Điện toán đám mây (Cloud Computing) có thể hiểu đơn giản: là các nguồn điện toán khổng lồ như Máy chủ, phần mềm, các dịch vụ,... sẽ nằm trên Internet thay vì trong máy tính cá nhân, máy tính gia đình và văn phòng để mọi người có thể kết nối và sử dụng bất cứ khi nào cần. Với các dịch vụ được cung cấp sẵn trên internet, doanh nghiệp sẽ không phải mua và duy trì hàng trăm, thậm chí hàng nghìn máy tính cũng như phần mềm. Các dịch vụ này có thể được mở rộng và thu hẹp tùy theo nhu cầu sử dụng của doanh nghiệp, và chi phí được tính theo mức độ sử dụng của khách hàng.

### 1.1.2. Các đặc tính cơ bản của điện toán đám mây

Điện toán đám mây có năm tính chất nổi bật so với mô hình truyền thống.[14]



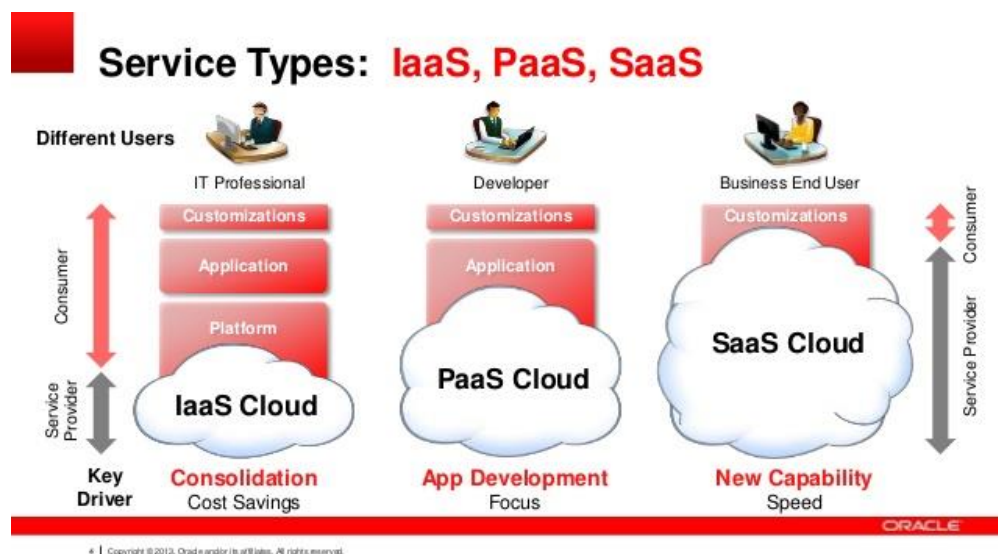
Hình 1.1: Mô hình điện toán đám mây

- ❖ **Tự phục vụ theo nhu cầu (On-demand self-service):** Người sử dụng dịch vụ có thể tự yêu cầu cung cấp các dịch vụ tài nguyên dưới dạng máy chủ, các dịch vụ phần mềm hay dịch vụ lưu trữ,...một cách tự động mà không cần phải qua nhà cung cấp dịch vụ.
- ❖ **Tính đàn hồi nhanh chóng (Rapid Elasticity):** Tài nguyên trên đám mây có thể được cung cấp một cách nhanh chóng và mềm dẻo. Có khả năng mở rộng hoặc thu hẹp theo nhu cầu hoặc theo tham số cấu hình. Có thể coi tài nguyên trên điện toán đám mây là không có giới hạn, và có thể được truy cập vào bất kỳ thời điểm nào.
- ❖ **Tập hợp tài nguyên (Resource pooling):** Tài nguyên máy tính của nhà cung cấp được gộp chung để phục vụ nhiều người dùng thông qua mô hình cho thuê. Các nguồn tài nguyên vật lý và ảo khác nhau được gán động và phân bổ lại theo nhu cầu của người dùng. Khách hàng không có quyền kiểm soát hoặc hiểu biết về vị trí chính xác của các tài nguyên được cung cấp nhưng có thể chỉ định ở mức trừu tượng cao (ví dụ như chỉ định quốc gia, vùng địa lý, trung tâm dữ liệu). Tài nguyên có thể bao gồm: lưu trữ, xử lý, bộ nhớ và băng thông mạng.

- ❖ **Truy cập mạng rộng rãi (Broad Network Access):** Dịch vụ đám mây luôn có sẵn sàng miễn là có kết nối internet. Chỉ cần từ 1 ứng dụng kết nối internet như máy tính để bàn, laptop, thiết bị di động,...là bạn đã có thể truy cập tới tài nguyên đám mây. Người dùng có thể truy cập mọi lúc, mọi nơi vào các dịch vụ đám mây.
- ❖ **Dịch vụ được đo đếm (Measured Service):** Hệ thống đám mây tự động kiểm soát và tối ưu hóa việc sử dụng tài nguyên bằng cách tận dụng khả năng đo lường đối với loại dịch vụ lưu trữ, xử lý, băng thông và tài khoản người dùng đang hoạt động. Khách hàng có thể theo dõi, kiểm tra các tài nguyên họ sử dụng, qua đó cung cấp sự minh bạch cho cả nhà cung cấp dịch vụ và khách hàng.

## 1.2. Các mô hình dịch vụ trong điện toán đám mây

Hiện nay có rất nhiều hãng công nghệ lớn nhỏ trên thế giới thực hiện cung cấp các dịch vụ điện toán đám mây với nhiều loại hình dịch vụ khác nhau. Nhưng có ba loại hình dịch vụ cơ bản là: Dịch vụ hạ tầng (IaaS – Infrastructure as a Service), dịch vụ nền tảng (PaaS – Platform as a Service), dịch vụ phần mềm (SaaS – Software as a Service). Cách phân loại này thường được gọi là “mô hình SPI”. [14]

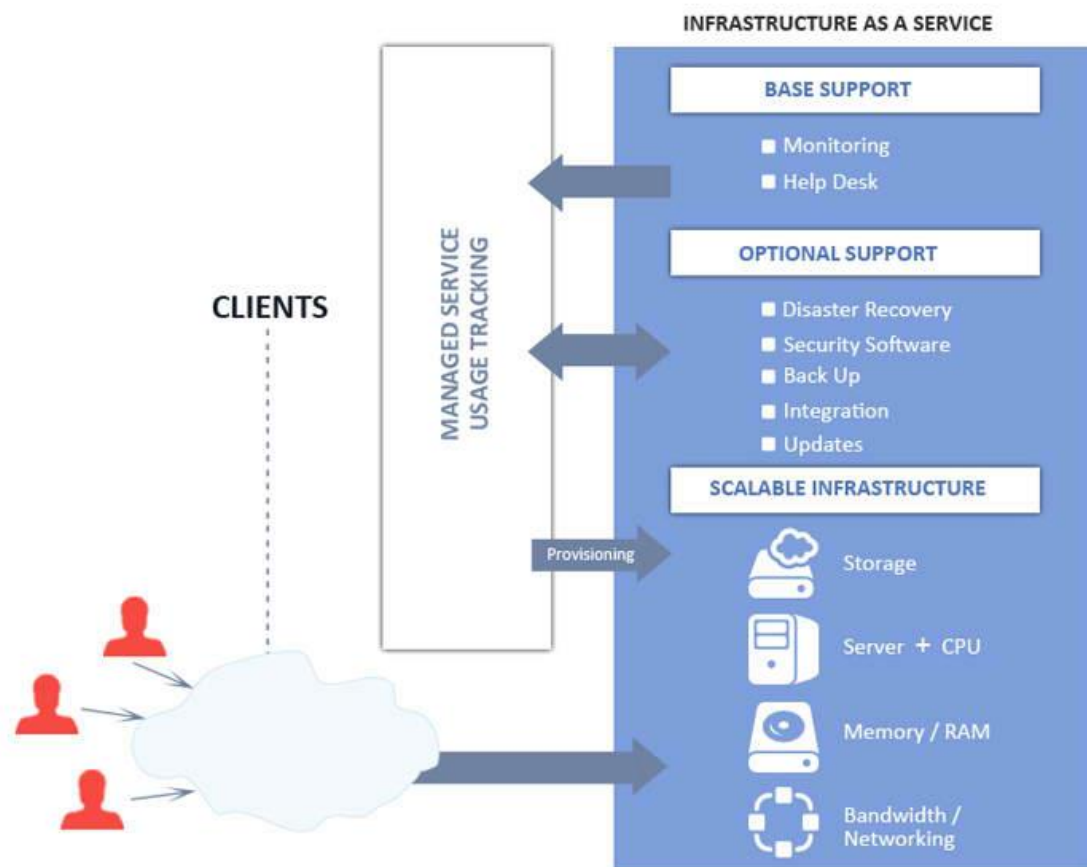


Hình 1.2: Các loại dịch vụ của ĐTĐM

### 1.2.1. Dịch vụ hạ tầng (IaaS – Infrastructure as a Service)

Dịch vụ IaaS là dịch vụ đám mây cung cấp cho người dùng hạ tầng thô (thường ở dưới hình thức các máy ảo) như một dịch vụ. Khách hàng có thể triển khai và chạy phần mềm trên các máy ảo như trên một máy chủ, hoặc có thể đưa dữ liệu cá nhân lên “đám

mây” và lưu trữ. Người dùng tuy không có quyền kiểm soát hạ tầng thực bên trong “đám mây”, nhưng họ có toàn quyền quản lý và sử dụng tài nguyên mà họ được cung cấp, cũng như yêu cầu mở rộng lượng tài nguyên họ được phép sử dụng.



Hình 1.3: Mô hình IaaS

Ví dụ điển hình về dịch vụ loại này là dịch vụ EC2 của Amazon hay Virtual Machine của Microsoft. Trên dịch vụ này, khách hàng có thể đăng ký sử dụng một máy tính ảo và lựa chọn một hệ điều hành (Windows, Ubuntu, CentOS ...) và tự cài đặt ứng dụng của mình trên đó

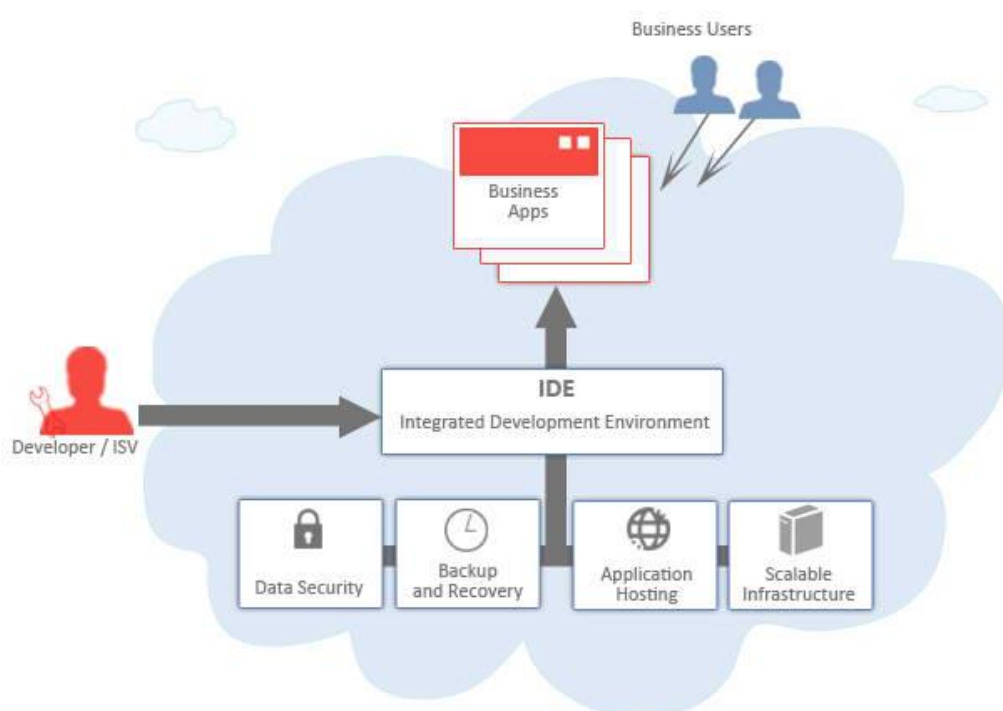
Các đặc trưng tiêu biểu [4]:

- ❖ Tài nguyên được cung cấp như là dịch vụ: bao gồm cả máy chủ, thiết bị mạng, bộ nhớ, CPU, không gian đĩa cứng, trang thiết bị trung tâm dữ liệu.
- ❖ Cung cấp khả năng linh hoạt trong việc mở rộng dịch vụ.
- ❖ Người sử dụng chỉ phải trả phí cho các tài nguyên thực tế sử dụng.
- ❖ Cho phép nhiều người dùng có thể cùng sử dụng trên một tài nguyên
- ❖ Đối với cấp độ doanh nghiệp: mang lại lợi ích cho công ty bởi một nguồn tài nguyên tính toán tổng hợp được cung cấp sẵn.



### 1.2.2. Dịch vụ nền tảng (PaaS – Platform as a Service)

Dịch vụ PaaS cung cấp nền tảng điện toán cho phép khách hàng phát triển, triển khai các phần mềm, phục vụ nhu cầu tính toán hoặc xây dựng thành dịch vụ trên nền tảng cloud đó. Dịch vụ PaaS hỗ trợ việc triển khai ứng dụng mà không quan tâm đến chi phí hay sự phức tạp của việc trang bị hạ tầng và quản lý các lớp phần cứng và phần mềm bên dưới. PaaS cung cấp tất cả các tính năng cần thiết để hỗ trợ việc xây dựng triển khai các ứng dụng và dịch vụ web trên Internet mà không cần bất kỳ thao tác tải hay cài đặt phần mềm nào. PaaS cho phép các nhà phát triển ứng dụng có thể tạo ra các ứng dụng một cách nhanh chóng do các rắc rối trong việc thiết lập máy chủ, cài đặt cơ sở dữ liệu đã được nhà cung cấp dịch vụ giải quyết.[14]



Hình 1.4: Mô hình PaaS

Ví dụ điển hình về mô hình PaaS trong thực tế là dịch vụ App Engine của Google. Dịch vụ App Engine cho phép khách hàng sử dụng ngôn ngữ lập trình Java hoặc Python để xây dựng các ứng dụng web chạy trên nền tảng App Engine. Hoặc một ví dụ khác là dịch vụ Web App service của Microsoft, dịch vụ Web App cũng cho phép khách hàng xây dựng các ứng dụng Web với môi trường chạy ứng dụng và phát triển trên ngôn ngữ .NET của Microsoft.

Các đặc trưng tiêu biểu [4]:

- ❖ Cung cấp môi trường phát triển tích hợp cho việc phát triển, kiểm thử, triển khai và vận hành ứng dụng.
- ❖ Khách hàng sử dụng công cụ trên nền web để khởi tạo, quản lý các dịch vụ.
- ❖ Kiến trúc của các dịch vụ được cung cấp trên cloud là đồng nhất.
- ❖ Tích hợp dịch vụ web và cơ sở dữ liệu.
- ❖ Hỗ trợ cộng tác nhóm phát triển.
- ❖ Cung cấp các công cụ hỗ trợ tiện tích khác.

### 1.2.3. Dịch vụ phần mềm (SaaS – Software as a Service)

Dịch vụ SaaS là một mô hình triển khai ứng dụng mà ở đó nhà cung cấp các dịch vụ là các phần mềm được triển khai trên internet. Khách hàng được phép sử dụng dịch vụ phần mềm này theo yêu cầu. Nhà cung cấp dịch vụ SaaS có thể lưu trữ ứng dụng trên máy chủ của họ hoặc tải ứng dụng xuống thiết bị khách hàng. Các dịch vụ này sẽ bị vô hiệu hóa sau khi thời hạn sử dụng kết thúc.[14]



Hình 1.5: Mô hình SaaS

Ví dụ điển hình về mô hình dịch vụ SaaS này phải kể đến Salesforce.com với các ứng dụng phục vụ cho doanh nghiệp mà nổi bật nhất là CRM. Các ứng dụng SaaS sử dụng cho người dùng cuối phổ biến là các ứng dụng Office Online của Microsoft hay Google Docs của Google.

Các đặc trưng tiêu biểu [4]:

- ❖ Là các phần mềm có sẵn và yêu cầu việc truy xuất, quản lý thông qua mạng internet.

- ❖ Quản lý các hoạt động một cách tập trung, cho phép khách hàng truy xuất từ xa thông qua web.
- ❖ Cung cấp ứng dụng thông thường, gắn gũi với mô hình ánh xạ từ một đến nhiều, bao gồm cả các đặc trưng kiến trúc, giá cả và quản lý.
- ❖ Tập trung phần mềm ở server của nhà cung cấp giúp giải phóng người dùng khỏi việc tải các bản vá lỗi và cập nhật.
- ❖ Thường xuyên tích hợp những phần mềm giao tiếp trên mạng diện rộng.

### 1.3. Các thành phần của điện toán đám mây

Điện toán đám mây gồm các thành phần cơ bản như dưới đây:

Khách hàng
Dịch vụ
Ứng dụng
Nền tảng
Lưu trữ
Hạ tầng

Hình 1.6: Thành phần của điện toán đám mây

- ❖ **Hạ tầng:** Cơ sở hạ tầng của điện toán đám mây là phần cứng được cung cấp như là các dịch vụ, nghĩa là các phần cứng này được chia sẻ và có thể sử dụng lại một cách dễ dàng. Các tài nguyên phần cứng này được cung cấp theo yêu cầu, điển hình như môi trường máy ảo. Dịch vụ kiểu này giúp cho việc giảm chi phí bảo hành, bảo trì, chi phí sử dụng ....Người dùng hoàn toàn có thể tắt các dịch vụ này trong trường hợp không sử dụng đến.
- ❖ **Lưu trữ:** Lưu trữ đám mây (Cloud Storage) gồm việc phân phối các dịch vụ lưu trữ dữ liệu như: cơ sở dữ liệu, dịch vụ web.
- ❖ **Nền tảng:** Cung cấp nền tảng cho điện toán và các giải pháp của dịch vụ, chi phối đến cấu trúc hạ tầng của “đám mây”. Đồng thời Platform cũng là điểm tựa cho lớp ứng dụng, cho phép các ứng dụng hoạt động trên nền tảng

đó. Platform giúp làm cho giảm sự tốn kém khi triển khai các ứng dụng do người dùng không phải trang bị cơ sở hạ tầng của riêng mình.

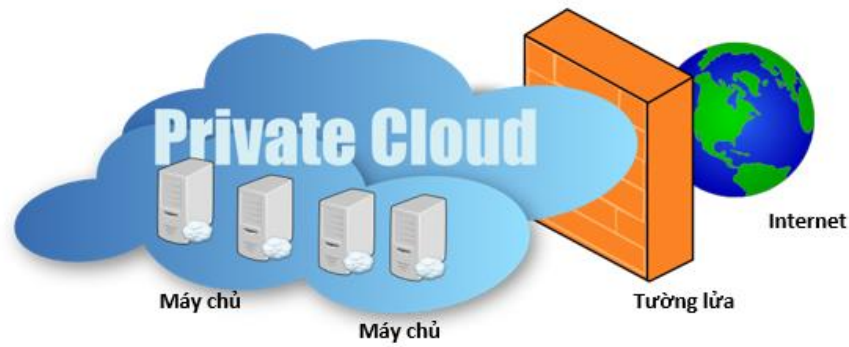
- ❖ **Ứng dụng:** Lớp ứng dụng của điện toán đám mây làm nhiệm vụ phân phối phần mềm như một dịch vụ thông qua mạng Internet. Qua đó khách hàng có thể truy cập và sử dụng các ứng dụng, dịch vụ mà không cần phải cài đặt trên máy của mình, các ứng dụng dễ dàng được chỉnh sửa và người sử dụng dễ dàng nhận được sự hỗ trợ.
- ❖ **Dịch vụ:** Dịch vụ đám mây là một phần độc lập của phần mềm có thể kết hợp với các dịch vụ khác để thực hiện tương tác, kết hợp giữa các máy tính khác nhau để thực hiện chương trình ứng dụng theo như yêu cầu. Ví dụ các dịch vụ hiện nay như Google Maps, các dịch vụ thanh toán linh hoạt trên Amazon, ....
- ❖ **Khách hàng:** Khách hàng đám mây bao gồm các phần cứng, phần mềm để dựa vào đó khách hàng có thể truy cập và sử dụng các ứng dụng hoặc dịch vụ được cung cấp từ đám mây. Ví dụ như máy tính và đường dây kết nối Internet (thiết bị phần cứng) và các trình duyệt Web (sản phẩm phần mềm) ....

## **1.4. Các mô hình triển khai điện toán đám mây**

### **1.4.1. Mô hình đám mây riêng (Private Cloud)**

Mô hình triển khai đám mây riêng (Private Cloud) là mô hình trong đó hạ tầng đám mây được sở hữu bởi một tổ chức hay doanh nghiệp và phục vụ cho người dùng của tổ chức hoặc doanh nghiệp đó. Những “đám mây” này tồn tại bên trong tường lửa của công ty và được các doanh nghiệp trực tiếp quản lý. [14]

Cơ sở hạ tầng đám mây riêng thường bao gồm các máy chủ, các thiết bị lưu trữ và các ứng dụng. Một hệ thống đám mây riêng được xây dựng tốt sẽ cung cấp khả năng bảo vệ an toàn dữ liệu và dịch vụ, cũng như tối ưu hóa các ứng dụng về độ sẵn sàng và hiệu năng sử dụng, nhằm cho phép các ứng dụng này đáp ứng nhanh chóng những nhu cầu mà người dùng cần, thích ứng với sự thay đổi và phát triển của công nghệ.

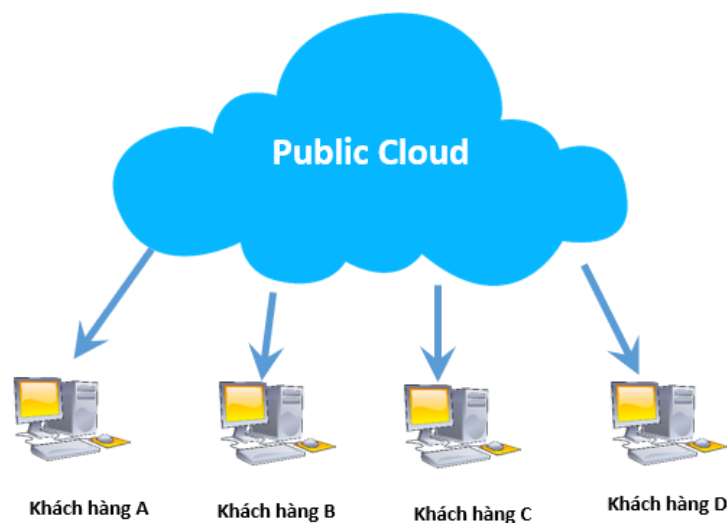


Hình 1.7: Mô hình đám mây riêng (Private Cloud)

Đám mây riêng được các tổ chức, doanh nghiệp lớn xây dựng nhằm khai thác ưu điểm về công nghệ và khả năng quản trị của điện toán đám mây. Các đám mây riêng giúp cho các doanh nghiệp tối ưu được hạ tầng của mình, nâng cao hiệu quả sử dụng, quản lý dễ dàng trong việc cấp phát và thu hồi tài nguyên, qua đó giảm thời gian đưa sản phẩm ra thị trường.

#### 1.4.2. Mô hình đám mây công (Public Cloud)

Mô hình đám mây công là các dịch vụ điện toán đám mây được các nhà cung cấp dịch vụ đưa ra cung cấp cho mọi người sử dụng rộng rãi thông qua hạ tầng Internet hoặc các mạng công cộng diện rộng. Các dịch vụ được cung cấp bởi nhà cung cấp dịch vụ và các ứng dụng của người dùng đều nằm trên đám mây. Các ứng dụng khác nhau chia sẻ chung tài nguyên tính toán, mạng và lưu trữ. [14]

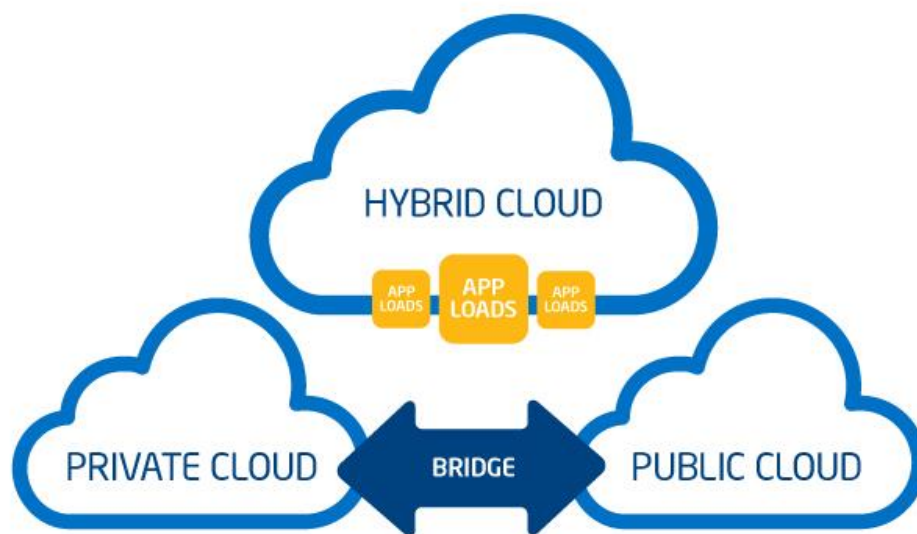


Hình 1.8: Mô hình đám mây công (Public Cloud)

Các dịch vụ đám mây công hướng tới số lượng khách hàng lớn nên thường có năng lực về hạ tầng cao, đáp ứng nhu cầu linh hoạt của khách hàng, qua đó giúp mang lại chi phí thấp cho khách hàng. Khách hàng sử dụng dịch vụ sẽ được lợi là chi phí đầu tư thấp, giảm thiểu rủi ro. Một lợi ích khác của mô hình này là cung cấp khả năng co giãn (mở rộng hoặc thu nhỏ) tùy theo yêu cầu của người sử dụng. Tuy nhiên, điện toán đám mây công có một trở ngại là tính bảo mật và tính an toàn dữ liệu. Trong mô hình này, mọi dữ liệu đều nằm trên dịch vụ Cloud do nhà cung cấp dịch vụ quản lý. Chính điều này làm cho các công ty lớn cảm thấy không an toàn đối với dữ liệu của mình khi sử dụng dịch vụ Cloud, đặc biệt là đối với các khách hàng hoạt động trong các lĩnh vực tài chính, ngân hàng hay các tổ chức chính phủ.

#### **1.4.3. Mô hình đám mây lai (Hybrid Cloud)**

Như đã phân tích ở trên, Public Cloud dễ dàng áp dụng, mức chi phí thấp nhưng không an toàn. Ở chiều ngược lại Private Cloud an toàn hơn, nhưng tốn chi phí và khó áp dụng. Do đó nếu kết hợp được hai mô hình này lại với nhau thì sẽ khai thác được những ưu điểm của từng mô hình. Đó chính là ý tưởng hình thành mô hình đám mây lai (Hybrid Cloud). [14]



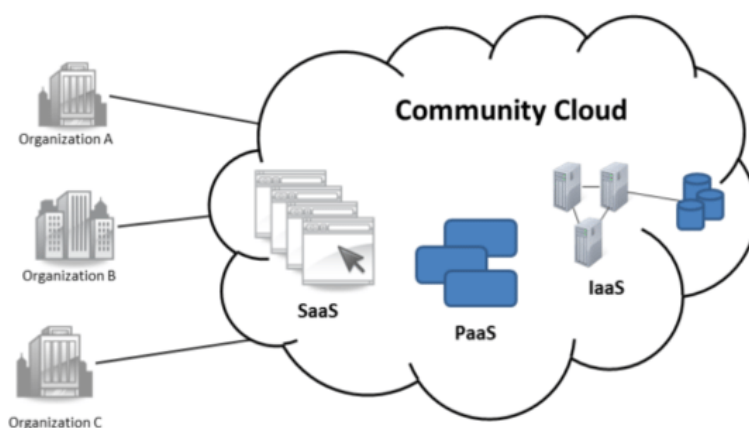
Hình 1.9: Mô hình kết hợp Hybrid Cloud

Hybrid Cloud là sự kết hợp của Public Cloud và Private Cloud, trong đó doanh nghiệp đưa các chức năng nghiệp vụ và dữ liệu không quan trọng lên đám mây, và sử dụng các dịch vụ Public Cloud để giải quyết và xử lý các dữ liệu này. Đồng thời đối với

các dữ liệu tương đối quan trọng, doanh nghiệp sẽ giữ chúng trong tầm kiểm soát bằng việc sử dụng Private Cloud.

#### 1.4.4. Mô hình đám mây cộng đồng (Community Cloud)

Đám mây cộng đồng là một mô hình triển khai điện toán đám mây, trong đó có nhiều doanh nghiệp được liên kết với nhau nhằm mục đích chia sẻ hạ tầng giữa các doanh nghiệp với nhau. Các doanh nghiệp này thường có chung lĩnh vực hoạt động kinh doanh, ngoài ra các doanh nghiệp này cũng phải có chung mối quan tâm về bảo mật, đảm bảo an toàn dữ liệu,...[14]



Hình 1.10: Mô hình đám mây cộng đồng

Khi triển khai mô hình điện toán đám mây cộng đồng, các doanh nghiệp có thể thực hiện quản lý đám mây theo nhiều cách khác nhau:

- ❖ Các doanh nghiệp có thể cùng nhau quản lý tài nguyên đám mây.
- ❖ Các doanh nghiệp có thể thuê bên thứ 3 đứng ra quản lý đám mây giúp cho họ
- ❖ Đám mây cộng đồng có thể liên quan tới nhiều tổ chức doanh nghiệp, các doanh nghiệp này có thể có nhiều chi nhánh khác nhau. Do đó, để quản lý có hiệu quả thì cần phải có người hoặc nhóm quản lý đám mây. Họ chịu trách nhiệm đơn đốc, kết hợp với các nhà quản lý chi nhánh để cùng tham gia quản lý đám mây.

### 1.5. Kết luận

Như vậy ở chương đầu tiên, luận văn đã mô tả những khái niệm cơ bản về điện toán đám mây, các đặc tính cơ bản của điện toán đám mây cũng như các mô hình dịch vụ được triển khai trên đám mây. Điện toán đám mây được triển khai dưới nhiều mô

hình khác nhau: đám mây riêng, đám mây công, đám mây lai và đám mây cộng đồng, hứa hẹn sẽ thu hút được sự quan tâm lớn từ cộng đồng.

Ở chương tiếp theo chúng ta sẽ đi vào tìm hiểu cụ thể về nền tảng điện toán đám mây Azure của Microsoft và các kiểu kiến trúc phần mềm triển khai trên nền tảng Microsoft Azure.



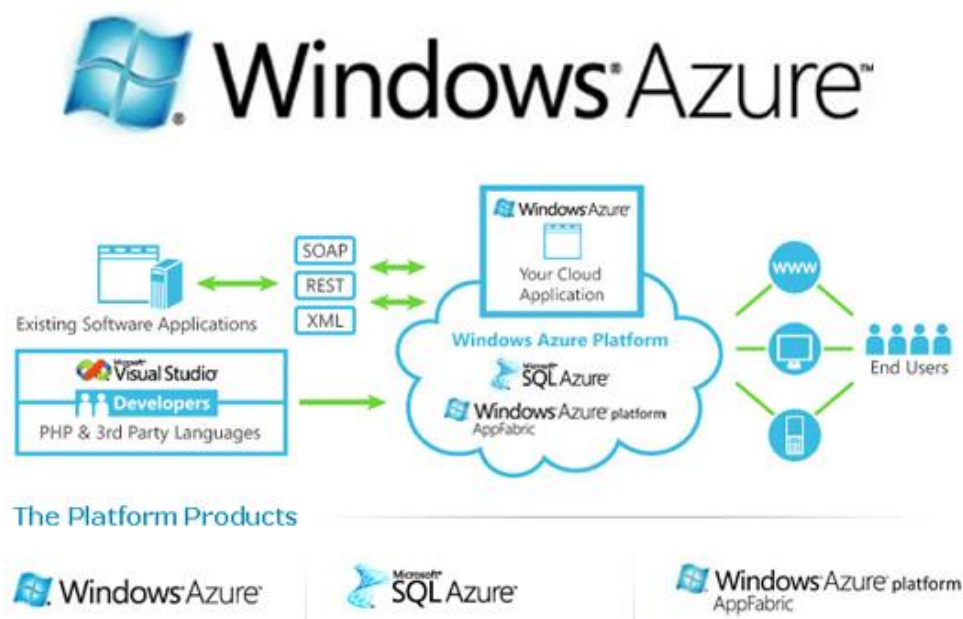
## Chương 2. KIẾN TRÚC PHẦN MỀM DỰA TRÊN CÁC DỊCH VỤ ĐIỆN TOÁN Đám Mây MICROSOFT AZURE

Ở phần này, luận văn trình bày tổng quan về các dịch vụ trên nền tảng Microsoft Azure và các hướng tiếp cận khi thiết kế kiến trúc phần mềm trên nền tảng điện toán đám mây Microsoft Azure.

### 2.1. Nền tảng Microsoft Azure

#### 2.1.1. Tổng quan về Window Azure Platform

Window Azure Platform cung cấp môi trường cho người sử dụng có thể dùng để phát triển và triển khai phần mềm trên nền tảng điện toán đám mây. Mỗi thành phần trong Window Azure được thiết kế nhằm cung cấp cho người dùng thực hiện một chức năng cụ thể của đám mây. Các thành phần chính trong nền tảng Window Azure bao gồm:



Hình 2.1: Tổng quan mô hình Azure Platform

(Nguồn: <https://www.thewindowsclub.com/secret-cloud-computing-microsoft>)

- **Windows Azure:** Cung cấp môi trường nền tảng, các hệ điều hành trên đám mây. Cung cấp nền tảng để chạy ứng dụng và lưu trữ dữ liệu trên máy chủ trong trung tâm dữ liệu của Microsoft.
- **SQL Azure:** Cung cấp dịch vụ lưu trữ dữ liệu quan hệ trên đám mây dựa trên SQL Server

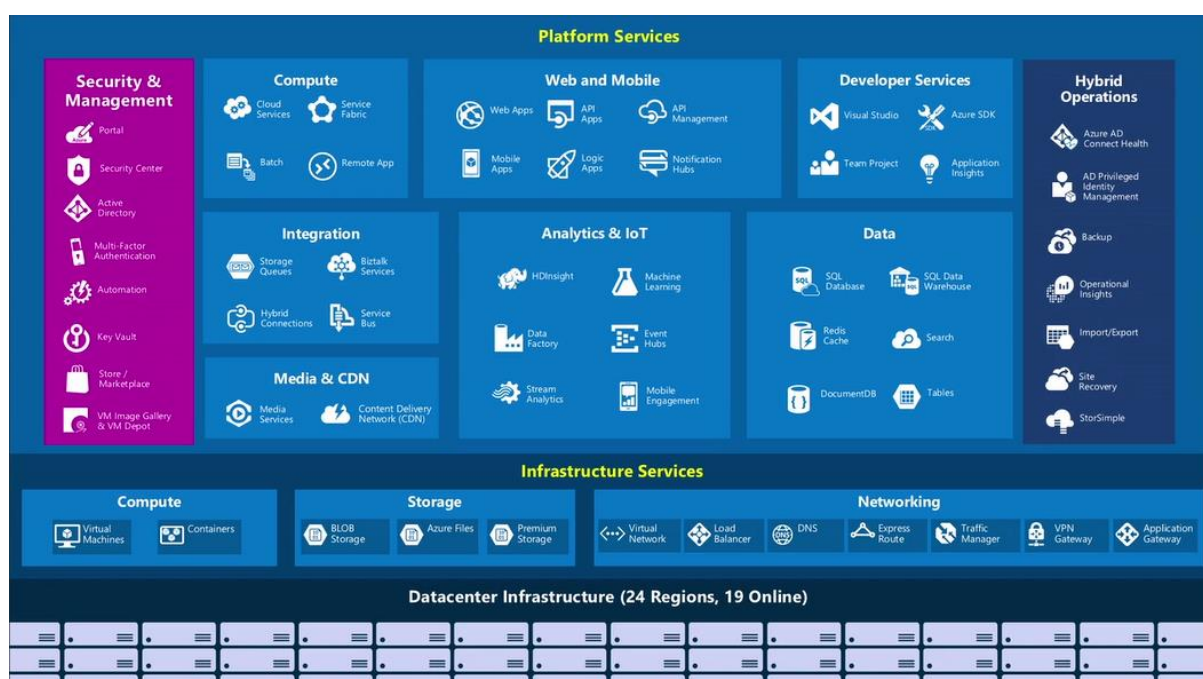
- **Windows Azure Platform AppFabric:** cung cấp các dịch vụ đám mây để kết nối các ứng dụng chạy trên đám mây hoặc on-premise.

### 2.1.2. Nền tảng Microsoft Azure

Microsoft Azure là một nền tảng điện toán đám mây do Microsoft cung cấp. Với Azure, mọi giới hạn về hạ tầng và địa lý bị phá bỏ. Người dùng có thể sử dụng nền tảng Azure theo nhiều cách khác nhau. Chẳng hạn, có thể sử dụng Microsoft Azure để xây dựng các ứng dụng web, hoặc lưu trữ dữ liệu trong Microsoft Azure Datacenters.[4]

Azure cung cấp rất nhiều dịch vụ khác nhau, cho phép bạn xây dựng, triển khai và quản lý cho hầu như mọi phải pháp công nghệ thông tin. Hay nói cách khác, Microsoft Azure là một thế giới của những khả năng không giới hạn.

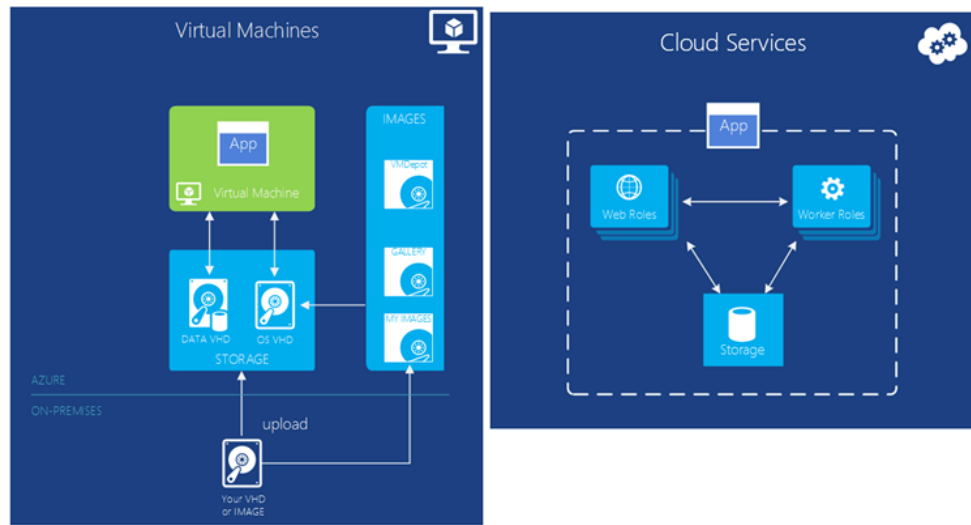
Để quản trị Microsoft Azure, Microsoft đã cung cấp cho chúng ta một giao diện portal để quản lý đó là Management Portal (<https://portal.azure.com>). Các dịch vụ hiện tại đang có trong Microsoft Azure được phân loại thành các nhóm dịch vụ khác nhau trong Management Portal. Mục đích chính của việc phân loại thành các nhóm dịch vụ trong Management Portal là giúp người dùng dễ dàng nhận ra và tiếp cận một cách nhanh chóng đến các dịch vụ đang được cung cấp trên Microsoft Azure.



Hình 2.2: Các thành phần của Microsoft Azure.

### a) Compute

Cung cấp các dịch vụ tính toán trên Azure, bao gồm các máy chủ ảo, các ứng dụng, và các dịch vụ trên nền đám mây.



Hình 2.3: Mô hình dịch vụ Compute của Microsoft Azure.

- **Virtual Machine:** Các máy chủ ảo này là phương thức sử dụng hạ tầng như một dịch vụ. Chúng ta có thể tạo ra một máy ảo với các hệ điều hành khác nhau như: Windows, Ubuntu, CentOS,... Các máy ảo này có thể được sử dụng theo rất nhiều cách khác nhau. Bạn có thể sử dụng chúng để tạo ra các môi trường phát triển và kiểm thử không quá đắt, khi mà bạn có thể tắt đi khi không sử dụng đến. Bạn có thể tạo và chạy các ứng dụng sử dụng bất kỳ ngôn ngữ lập trình hay thư viện nào mà bạn muốn.
- **Cloud services:** Cho phép bạn xây dựng và triển khai các ứng dụng đảm bảo tính sẵn sàng cao và có khả năng mở rộng với hầu hết các ngôn ngữ lập trình. Chúng ta có thể sử dụng các công nghệ: .NET, PHP, Node.JS, Java, Python và các cơ sở dữ liệu như SQL Server, MySQL, Oracle.
- **Service Fabric:** Service Fabric Platform là một thế hệ tiếp theo của Platform-as-a-Service được cung cấp bởi microsoft. Nó được dùng thử từ tháng 5 năm 2015 và chính thức vào đầu năm 2016. Nó không chỉ hỗ trợ stateless services mà còn hỗ trợ stateful services để xây dựng một ứng dụng doanh nghiệp có hiệu quả cao, tin cậy và có khả năng mở rộng.

### b) Data management

Các ứng dụng cần phải có dữ liệu, và các ứng dụng khác nhau sẽ đòi hỏi các loại dữ liệu khác nhau. Vì lý do đó, Azure cung cấp nhiều cách khác nhau để lưu trữ

và quản lý dữ liệu như SQL Database, dữ liệu không có cấu trúc (Blobs), dữ liệu dạng bảng (Tables).

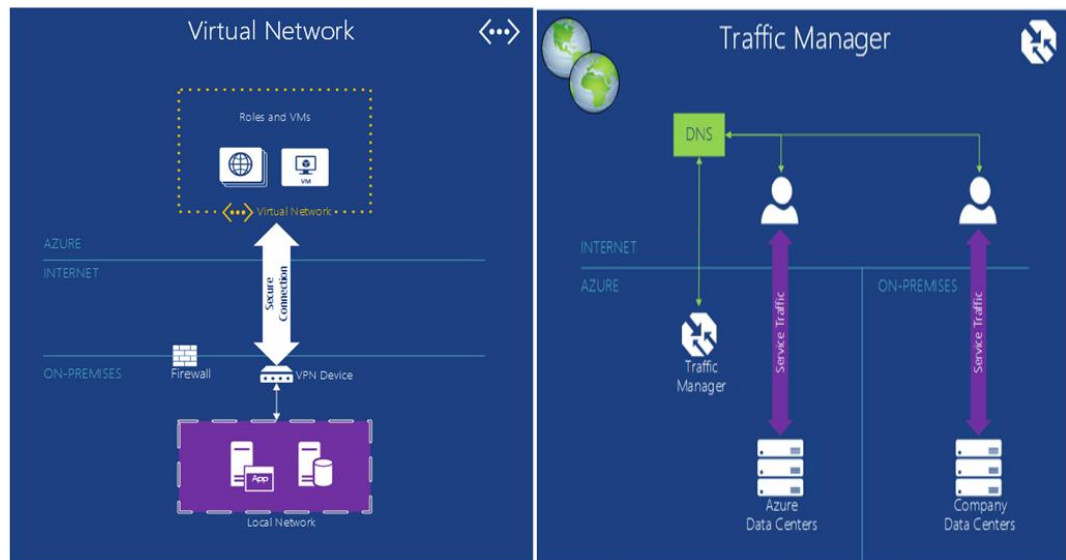


Hình 2.4: Dữ liệu được quản lý trong Microsoft Azure.

- **SQL Database:** Được sử dụng cho việc lưu trữ dữ liệu quan hệ. SQL Database cung cấp tất cả các tính năng chủ chốt của một hệ quản trị cơ sở dữ liệu quan hệ, bao gồm các giao dịch, truy cập dữ liệu đồng thời bởi nhiều người sử dụng, tính toàn vẹn của dữ liệu,... và các mô hình lập trình quen thuộc. SQL Database có thể được truy cập sử dụng bằng Entity Framework, ADO.NET, JDBC, và các công nghệ truy cập dữ liệu quen thuộc khác. Nó cũng hỗ trợ hầu hết ngôn ngữ T-SQL, đi kèm với SQL Server Tools như SQL Server Management Studio.
- **Tables:** Tables không được sử dụng để cung cấp giải pháp lưu trữ quan hệ. Trong thực tế nó là một ví dụ của cách tiếp cận NoSQL được gọi là lưu trữ cặp khoá/giá trị (key/value). Thay vì Windows Azure cho phép ứng dụng lưu trữ các thuộc tính của vài kiểu khác nhau như: chuỗi ký tự, số, và ngày tháng. Một ứng dụng có thể lấy ra một nhóm các thuộc tính này bằng việc cung cấp một khoá duy nhất cho nhóm đó. Trong khi đó các toán tử phức tạp như phép join không được hỗ trợ, Tables cho phép truy cập nhanh tới các dữ liệu đã được định kiểu. Do đó chúng rất dễ mở rộng, với một table có khả năng chứa hàng terabyte dữ liệu. Và dễ dàng lôi dữ liệu ra, do đó Tables sẽ rẻ hơn rất nhiều so với việc lưu trữ dữ liệu quan hệ sử dụng SQL Database.
- **Blobs:** Lựa chọn thứ 3 cho việc quản lý dữ liệu, Windows Azure Blobs, được thiết kế để lưu trữ dữ liệu nhị phân không cấu trúc. Giống như Tables, Blobs cung cấp giải pháp lưu trữ không đắt, và một blob có thể chứa hàng terabyte. Ví dụ một ứng dụng mà lưu trữ Video, hoặc sao lưu dữ liệu hoặc thông tin nhị phân có thể sử dụng Blobs cho đơn giản với mức chi phí khá rẻ.

### c) Networking

Cung cấp các tùy chọn khác nhau để kết nối người dùng với trung tâm dữ liệu, để tạo nên kiến trúc mạng trên nền Azure.

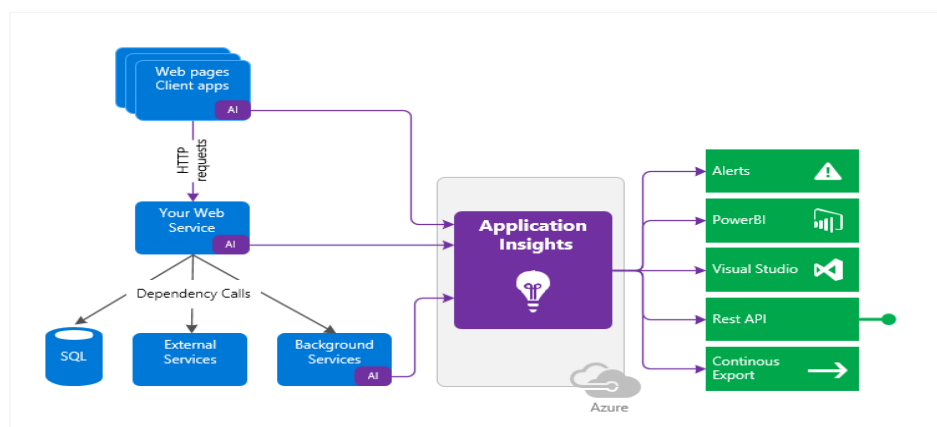


Hình 2.5: Mô hình Networking trên Azure.

- **Virtual Network:** Azure Virtual Network là chức năng mạng ảo trong Microsoft Azure. Máy ảo và dịch vụ là một phần của mạng ảo và chúng có thể truy cập qua lại với nhau.
- **Traffic manager:** Azure Traffic Manager cho phép bạn kiểm soát được việc phân phối truy cập của người dùng tới các endpoint của hệ thống đặt ở nhiều datacenter khác nhau dựa vào các phương thức điều hướng truy cập được Azure Traffic Manager cung cấp cũng như dựa vào tình trạng “sức khỏe” của các endpoint.

#### d) Developer services

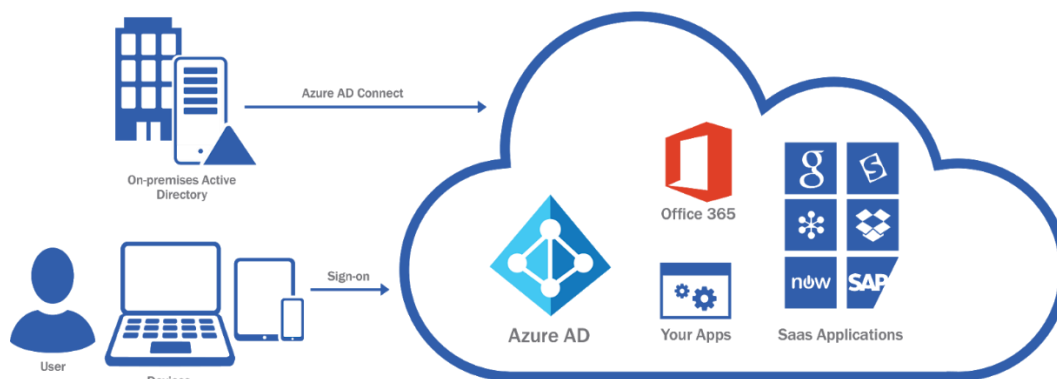
Cung cấp các công cụ để lập trình viên có thể xây dựng và triển khai các ứng dụng trên nền tảng Azure. Bao gồm các công cụ như: Visual Studio Online, Azure SDK, Azure tools for Visual Studio, Automation và APIs



Hình 2.6: Mô hình Application Insight trên Azure.

### e) Security & Management

Đơn giản hóa việc quản trị và bảo mật ứng dụng doanh nghiệp cho nhân viên IT bằng cách bảo mật quyền truy cập cùng Multi-Factor Authentication và Self-service reset password. Với Azure Active Directory, Microsoft cung cấp nhiều gói dịch vụ khác nhau nhằm đáp ứng những nhu cầu khác nhau của khách hàng



Hình 2.7: Mô hình Active Directory trên Azure.

### f) Web & Mobile

- **Web:** Windows Azure cung cấp một nền tảng mạnh mẽ và an toàn cho website doanh nghiệp. Bạn có thể sử dụng Windows Azure Active Directory để xác thực, kiểm soát truy cập, an toàn, bạn có thể lưu trữ dữ liệu kinh doanh trang web của bạn trong cơ sở dữ liệu. Bạn có thể tạo các trang web của bạn bằng cách sử dụng ngôn ngữ tùy chọn, chẳng hạn như ASP.NET, PHP, Node.js, Python. Và nếu muốn nhanh hơn, bạn có thể nhanh chóng xây dựng trang web của bạn bằng cách sử dụng một khuôn mẫu phổ biến hoặc mẫu từ Azure App Gallery Windows, trong đó bao gồm WordPress, Umbraco, DotNetNuke, Drupal, Django, CakePHP, và Express.
- **Mobile:** Windows Azure cho phép bạn xây dựng và triển khai một giải pháp đám mây back-end cho các ứng dụng mobile. Bạn có thể sử dụng nền tảng phát triển phổ biến như .NET hoặc NodeJS để tạo ra giải pháp, sau đó triển khai nó đến các đám mây sử dụng Windows Azure Virtual Machines, dịch vụ đám mây, hoặc dịch vụ di động. Windows Azure Mobile Services hỗ trợ đa nền tảng cho việc phát triển các giải pháp cho hầu hết mọi nền tảng bao gồm Windows Phone, Windows Store, Android, Apple iOS, và HTML5. Windows Azure Hubs Notification cho phép bạn đẩy thông báo cho người sử dụng để cho phép ứng dụng tương tác với người dùng theo thời gian thực, và bạn có thể sử dụng các nền tảng truyền thông xã hội từ Microsoft, Google, Facebook, Twitter hoặc cho các mục đích xác thực người dùng.



### g) Backup

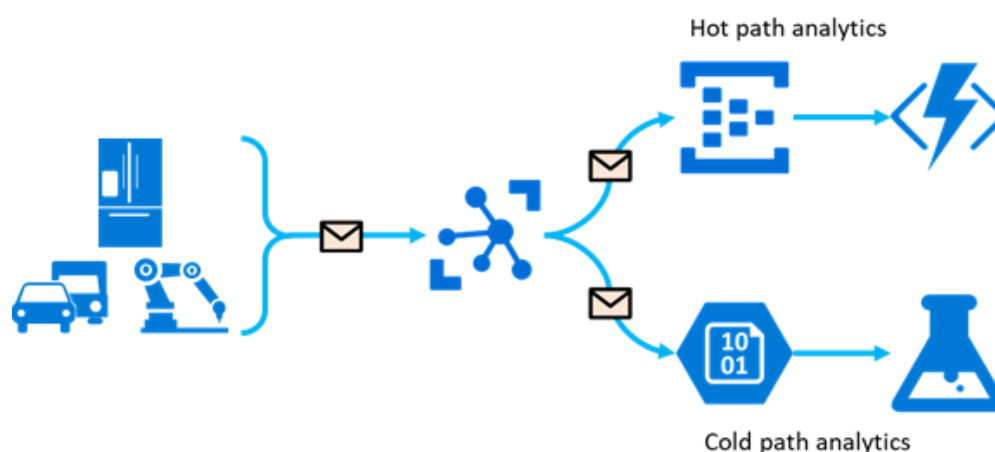
Azure cung cấp các dịch vụ sao lưu dữ liệu, ứng dụng hoạt động trên nền tảng Azure một cách tự động. Microsoft cung cấp dịch vụ Azure Backup để sao lưu dữ liệu và giúp khách hàng phục hồi dữ liệu khi có sự cố xảy ra. Tùy vào gói dịch vụ, Microsoft sẽ nhân bản backup thành 3 hoặc 9 bản giống nhau và đặt ở những máy chủ tách biệt nhằm đảm bảo khả năng phục hồi cho khách hàng lên đến 99,9%.

### h) Integration

Windows Azure cung cấp một số tùy chọn khác nhau cho việc tích hợp cơ sở hạ tầng hiện có tại chỗ của bạn với các ứng dụng của bạn đang chạy trong đám mây công cộng Windows Azure. Windows Azure Service Bus có thể được sử dụng cho giao tiếp giữa on-premise, các ứng dụng dựa trên đám mây và dịch vụ

### i) Analytics & IoT

Microsoft hiện đang cung cấp 2 bộ IoT trên Azure là Azure IoT Hub và Azure IoT Suite. Azure IoT Hub cung cấp cho bạn các công cụ phát triển (SDK) để đưa telemetry data lên Azure, cũng như quản lý việc xác thực, các event hoặc cung cấp back-end để quản lý thiết bị IoT. Bộ Azure IoT Hub này cũng cung cấp các SDK để phát triển trên IoT Gateway, các thư viện liên quan đến xác thực người dùng, protocol (HTTP, MQTT, ZIGBEE...)

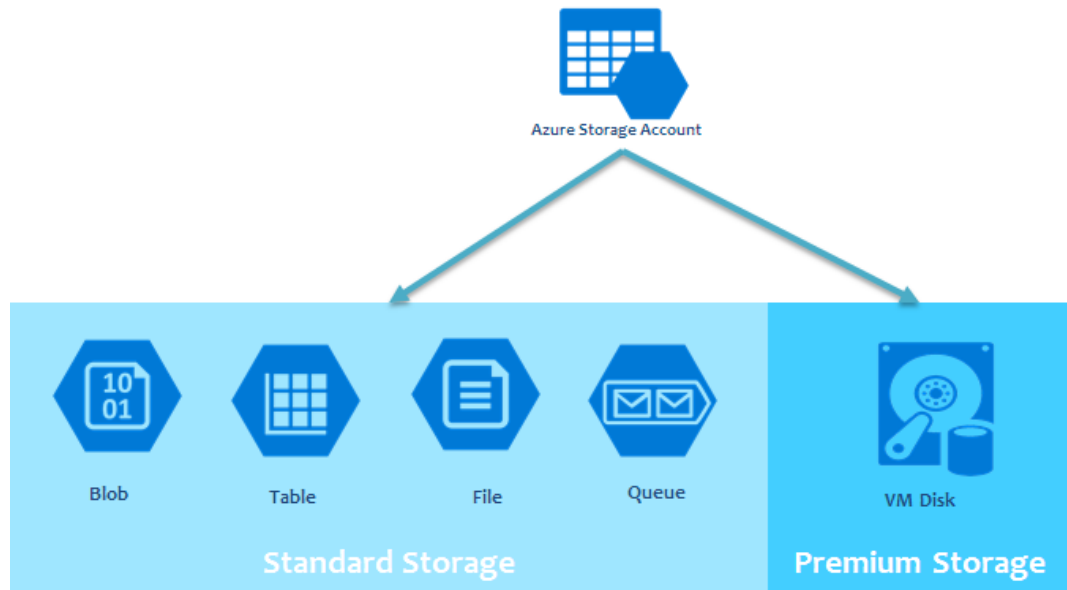


Hình 2.8: Mô hình sử dụng IoT Hub trên Azure.

### j) Storage

Azure Blob Storage là một dịch vụ hay đơn giản là một công cụ cho phép lưu trữ dữ liệu không cấu trúc trên cloud. Mỗi dữ liệu đưa lên để lưu trữ thì ta coi đó như một đối tượng, có thể là dữ liệu văn bản, dữ liệu nhị phân, các tài liệu hay media

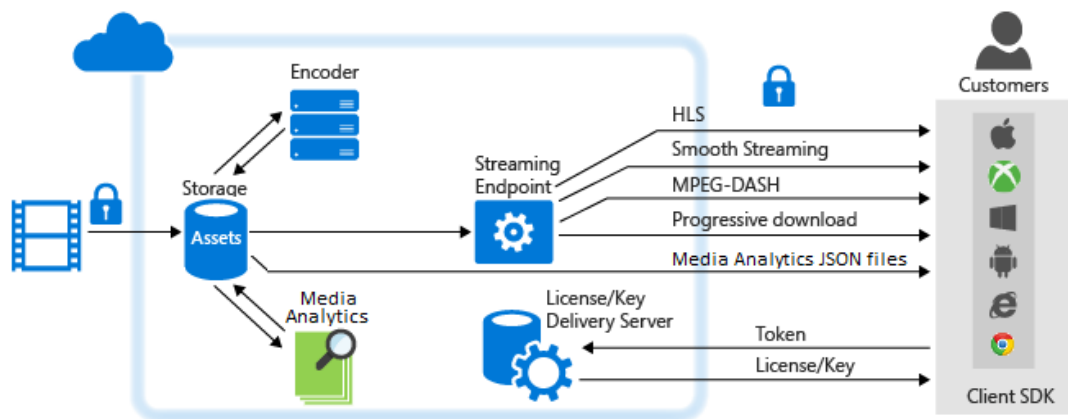
file, hoặc là các file cài đặt ... Blob storage hay còn được gọi là Object storage. Azure Blob Storage là NON-SQL Database.



Hình 2.9: Mô hình của Media Service.

#### k) Media

Windows Azure Media Services giúp dễ dàng để cung cấp cho doanh nghiệp một sự hiện diện trên phương tiện truyền thông toàn cầu. Bạn có thể nhanh chóng xây dựng quy trình làm việc end-to-end sử dụng dịch vụ từ cả Microsoft và các đối tác của mình. Phương tiện truyền thông của bạn có thể được bảo vệ bằng Digital Rights Management (DRM), và Advanced Encryption Standard (AES) hoặc PlayReady được sử dụng để bảo vệ nó trong quá trình phát lại



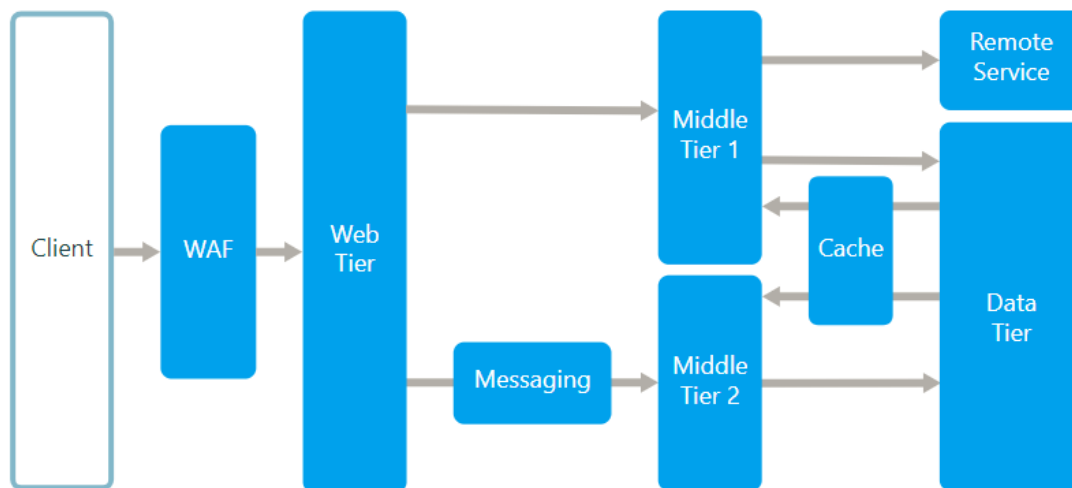
Hình 2.10: Mô hình của Media Service.



## 2.2. Các kiểu kiến trúc phần mềm trên Cloud

### 2.2.1. Kiến trúc phân tầng (N-tier)

#### a) Tổng quan



Hình 2.11: Mô hình Kiến trúc phân tầng [11]

Kiến trúc phân tầng (N-tier) thực hiện phân chia ứng dụng thành các tầng logic và các tầng vật lý. Chia tầng là một cách để phân tách trách nhiệm và quản lý các phụ thuộc. Mỗi tầng có trách nhiệm cụ thể. Một lớp ở tầng trên có thể sử dụng cách dịch vụ trong một lớp ở tầng thấp hơn.

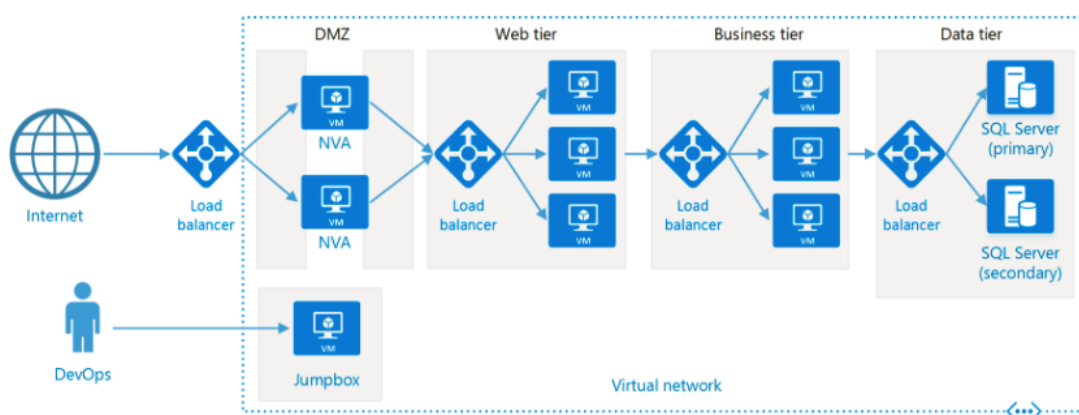
Các tầng được tách riêng về mặt vật lý, được chạy trên các máy riêng biệt. Một tầng có thể gọi trực tiếp đến một tầng khác hoặc sử dụng các tin nhắn không đồng bộ (asynchronous messaging) hoặc sử dụng hàng đợi tin nhắn. Việc phân tách các tầng vật lý giúp cải thiện khả năng mở rộng và khả năng phục hồi. Nhưng việc này cũng làm tăng thêm độ trễ do việc giao tiếp qua mạng.

Thông thường một ứng dụng phân tầng thường có ba tầng là tầng trình diễn, tầng giữa và tầng cơ sở dữ liệu. Các ứng dụng phức tạp hơn thì có thể có nhiều hơn ba tầng.

Các lớp trong một ứng dụng phân tầng có thể được xây dựng dưới dạng phân lớp đóng hoặc phân lớp mở:

- *Kiến trúc phân lớp đóng*: Mỗi tầng chỉ có thể gọi tới tầng ngay phía dưới của nó.
- *Kiến trúc phân lớp mở*: Mỗi tầng chỉ có thể gọi tới bất kỳ tầng nào ở phía dưới của nó.

## b) Mô hình kiến trúc phân tầng trên Azure



Hình 2.12: Mô hình Kiến trúc phân tầng trên Azure [11]

Mỗi tầng bao gồm hai hoặc nhiều máy ảo. Nhiều máy ảo cung cấp khả năng phục hồi trong trường hợp một máy ảo bị lỗi. Cân bằng tải được sử dụng để phân phối các yêu cầu bên trên các máy ảo trong mỗi tầng. Một tầng có thể mở rộng theo chiều ngang bằng cách thêm nhiều máy ảo hơn vào nhóm.

Mỗi tầng cũng được đặt bên trong mạng con riêng của nó, có nghĩa là địa chỉ IP nội bộ của chúng nằm trong cùng một dải địa chỉ. Điều đó giúp dễ dàng áp dụng các quy tắc nhóm bảo mật mạng (NSG) và các bảng định tuyến cho các tầng riêng lẻ.

Ứng dụng Web và các tầng nghiệp vụ là phi trạng thái. Bất kỳ máy ảo nào cũng có thể tiếp nhận yêu cầu xử lý cho tầng đó. Tầng dữ liệu nên bao gồm các dữ liệu có thể nhân rộng được. Đối với Windows, thì nên sử dụng dịch vụ SQL Server.

Nhóm bảo mật mạng (NSG) hạn chế quyền truy cập vào từng tầng, ví dụ tầng cơ sở dữ liệu chỉ cho phép truy cập từ tầng nghiệp vụ.

### Các lưu ý:

- Kiến trúc N-tier không bị giới hạn ở ba tầng, đối với các ứng dụng phức tạp hơn thì có thể có nhiều tầng hơn.
- Mỗi tầng có các yêu cầu riêng về tính mở rộng, tính sẵn sàng và tính bảo mật.
- Xem xét trong kiến trúc xem có thể sử dụng các dịch vụ có sẵn mà không ảnh hưởng nhiều đến kiến trúc. Các dịch vụ có thể cân nhắc sử dụng gồm bộ nhớ đệm, tin nhắn, lưu trữ và cơ sở dữ liệu
- Để tăng tính bảo mật, hãy thêm một mạng DMZ ở phía trước của ứng dụng. DMZ bao gồm các thiết bị mạng ảo (Network Virtual Appliances – NVA) thực hiện các chức năng bảo mật như tường lửa và kiểm tra gói tin

- Để tăng tính sẵn sàng, có thể đặt hai hoặc nhiều NVA với bộ cân bằng tải ngoài để phân phối các yêu cầu trên mạng Internet tới các instances.
- Không cho phép truy cập từ xa RDP hoặc SSH trực tiếp vào máy ảo đang cài đặt ứng dụng. Thay vào đó, sử dụng một máy ảo khác làm jumpbox để kết nối với các máy ảo khác. Jumpbox có NSG cho phép RDP hoặc SSH từ các địa chỉ IP được cấu hình từ trước.

### c) Kiến trúc phân tầng được sử dụng khi nào

Kiến trúc phân tầng thường được triển khai dưới dạng các dịch vụ hạ tầng (IaaS), với mỗi tầng sẽ được chạy trên mộ máy ảo riêng biệt. Tuy nhiên, một ứng dụng phân tầng không nhất thiết phải sử dụng hoàn toàn các dịch vụ IaaS thuần túy, mà có thể được áp dụng với các dịch vụ khác như bộ nhớ đệm, dịch vụ tin nhắn, dịch vụ lưu trữ dữ liệu.

Kiến trúc phân tầng thường được áp dụng cho các trường hợp sau:

- Các ứng dụng web đơn giản
- Thực hiện chuyển đổi một ứng dụng on-premise lên Azure với điều kiện việc chỉnh sửa, tái cấu trúc là tối thiểu
- Phát triển thống nhất các ứng dụng on-premise và trên đám mây

Kiến trúc phân tầng rất phổ biến trong các ứng dụng truyền thống on-premise, do đó nó là một sự phù hợp tự nhiên để chuyển đổi hệ thống lên Azure.

### d) Ưu điểm

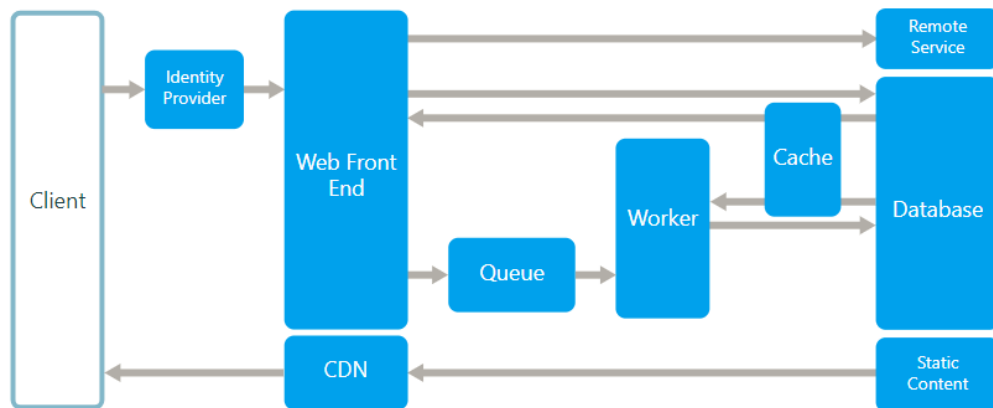
- Khả năng chuyển đổi từ server tại chỗ lên server cloud một cách đơn giản do tính tương đồng giữa 2 môi trường
- Thời gian học tập của các lập trình viên giảm do chủ yếu sử dụng các máy ảo.
- Là sự tiến hóa tự nhiên từ mô hình ứng dụng truyền thống
- Tăng tính mở cho môi trường không đồng nhất (Windows/ Linux)

### e) Hạn chế

- Việc thiết kế nguyên khối làm ngăn cản việc triển khai các tính năng độc lập
- Quản lý ứng dụng IaaS hoạt động hiệu quả hơn là việc ứng dụng chỉ sử dụng các dịch vụ được quản lý sẵn.
- Khó khăn trong việc quản lý bảo mật mạng đối với các hệ thống lớn.

## 2.2.2. Kiến trúc Web - Queue - Worker

### a) Tổng quan



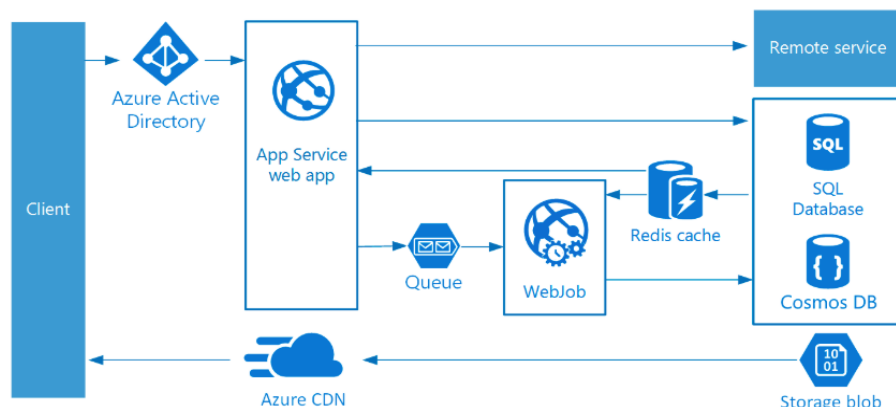
Hình 2.13: Mô hình Kiến trúc Web – Queue – Worker [11]

Thành phần chính của kiến trúc này là một giao diện web thực hiện tiếp nhận các yêu cầu của người dùng và một worker thực hiện các tác vụ tiêu tốn nhiều tài nguyên, các công việc theo luồng hoặc các công việc theo lô. Giao diện web thực hiện giao tiếp với worker thông qua các thông điệp được gửi vào hàng đợi.

Các thành phần khác thường được tích hợp vào kiến trúc này bao gồm:

- Một hoặc nhiều cơ sở dữ liệu
- Bộ nhớ đệm lưu trữ các giá trị từ cơ sở dữ liệu nhằm mục đích tăng tốc độ đọc.
- CDN để phục vụ nhanh các nội dung tĩnh
- Các dịch vụ từ xa như dịch vụ gửi SMS, dịch vụ Email
- Các dịch vụ xác thực.

### b) Mô hình kiến trúc trên Azure



Hình 2.14: Mô hình Kiến trúc Web – Queue - Worker trên Azure [11]

Ứng dụng web được triển khai trên Azure App Service, còn các worker được triển khai trên các WebJob.

Hàng đợi dùng lưu trữ các thông điệp thì ta có thể sử dụng Azure Service Bus hoặc Azure Storage Queue

Azure Redis Cache được sử dụng để lưu trữ dữ liệu trong bộ nhớ đệm, giúp giảm thời gian truy cập dữ liệu

Azure CDN được sử dụng để làm bộ nhớ đệm cho các dữ liệu tĩnh như hình ảnh, Css hoặc các file HTML

Để lưu trữ dữ liệu, có thể chọn những công nghệ phù hợp với từng ứng dụng, có thể là SQL Database, Cosmos DB, MySQL, MongoDB,... Để minh họa thì hình vẽ trên có sử dụng SQL Database và Cosmos DB.

#### Các lưu ý:

- Không phải mọi giao dịch đều phải thực hiện qua hàng đợi và worker để lưu trữ. Ứng dụng web có thể thực hiện các thao tác đọc, ghi đơn giản trực tiếp. Các worker được thiết kế cho các tác vụ tiêu tốn nhiều tài nguyên hệ thống hoặc thời gian thực hiện xử lý lâu.
- Sử dụng các tính năng tự động đã được tích hợp sẵn của App Service để thực hiện mở rộng số lượng instances. Nếu khả năng chịu tải của ứng dụng có thể dự đoán trước được thì có thể sử dụng chế độ lập lịch tự động. Nếu khả năng chịu tải là không thể dự đoán trước, hãy sử dụng chế độ tự động dựa trên số liệu tính toán tài nguyên của hệ thống.
- Có thể cân nhắc đưa Web App và WebJob vào các gói dịch vụ riêng biệt, bằng cách đó chúng có thể được mở rộng hoặc thu nhỏ một cách độc lập với nhau. Qua đó tối ưu được chi phí sử dụng
- Phân chia thành các môi trường độc lập: môi trường phát triển, môi trường kiểm thử, và môi trường sản phẩm.

#### **c) Kiến trúc được sử dụng khi nào**

Kiến trúc Web – Queue – Worker thường được sử dụng trong một số trường hợp sau:

- Các ứng dụng có miền tương đối đơn giản
- Ứng dụng có một số quy trình công việc hoạt động mất nhiều thời gian

- Khi muốn sử dụng các dịch vụ được quản lý thay vì sử dụng các cơ sở hạ tầng của Azure (IaaS).

#### d) Ưu điểm

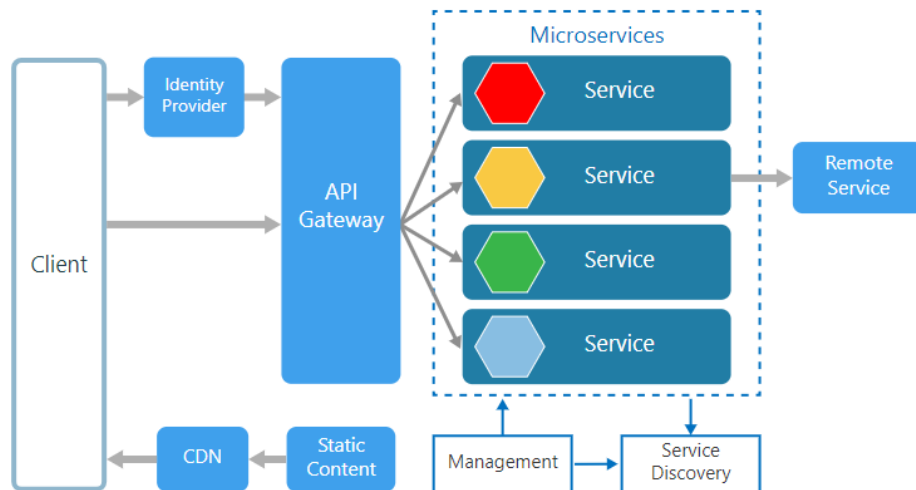
- Kiến trúc tương đối đơn giản, dễ hiểu
- Dễ triển khai và quản lý
- Phân tách rõ ràng các mối quan tâm
- Giao diện người dùng được tách riêng khỏi worker bằng cách sử dụng các thông điệp bất đồng bộ.
- Ứng dụng web và worker có thể mở rộng hoặc thu hẹp một cách độc lập với nhau.

#### e) Hạn chế

- Nếu không thiết kế cẩn thận, ứng dụng web và worker có thể trở thành các thành phần khó duy trì và cập nhật
- Có thể có các phụ thuộc ẩn, nếu ứng dụng web và worker chỉ sử dụng dữ liệu và các mô đun mã nguồn.

### 2.2.3. Kiến trúc vi dịch vụ (Microservice)

#### a) Tổng quan



Hình 2.15: Mô hình Kiến trúc Microservice [11]

Các đặc tính của kiến trúc vi dịch vụ:

- Trong kiến trúc vi dịch vụ, các dịch vụ được phân chia thành các thành phần nhỏ, độc lập và được kết hợp một cách lỏng lẻo
- Mỗi dịch vụ là một mã nguồn độc lập, có thể được quản lý bởi một nhóm nhỏ các lập trình viên

- Các dịch vụ có thể được triển khai một cách độc lập. Một nhóm có thể nâng cấp một dịch vụ đã có mà không cần phải triển khai lại toàn bộ ứng dụng.
- Mỗi dịch vụ chịu trách nhiệm lưu trữ dữ liệu của riêng mình.
- Các dịch vụ giao tiếp với nhau bằng cách sử dụng các API đã được xác định rõ. Chi tiết thực thi của từng dịch vụ được che dấu đối với các dịch vụ khác.
- Các dịch vụ có thể không cần phải sử dụng chung một công nghệ, thư viện hoặc các framework.

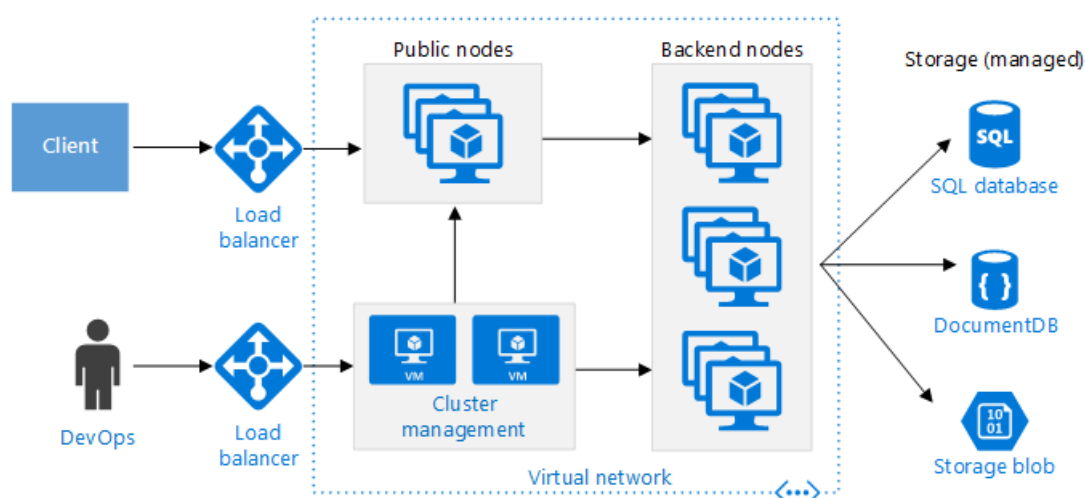
Bên cạnh các thành phần chính của microservice, thì còn có các thành phần khác:

**Management:** Là thành phần quản lý chịu trách nhiệm đặt các dịch vụ trên các nút, xác định lỗi và các dịch vụ cân bằng tải trên nút.

**Service Discovery:** Duy trì một danh sách các dịch vụ và các nút mà các dịch vụ đang nằm trên đó.

**API Gateway:** Là điểm đầu vào cho client. Client không trực tiếp gọi tới các dịch vụ, thay vào đó nó sẽ gọi đến API Gateway, tại đây API Gateway sẽ chuyển tiếp yêu cầu tới dịch vụ phù hợp.

## b) Mô hình kiến trúc trên Azure



Hình 2.16: Mô hình Kiến trúc Microservice sử dụng Azure Container Service [11]

**Public nodes:** Các nút này có thể được truy cập thông qua bộ cân bằng tải. API Gateway sẽ được lưu trữ trên các nút này.

**Backend nodes:** Các nút này chạy các dịch vụ mà khách hàng muốn thông qua API Gateway. Các nút này không nhận được lưu lượng truy cập internet trực tiếp. Các nút này có thể bao gồm nhiều nhóm máy ảo với các cấu hình phần cứng có thể khác nhau.

**Cluster management:** Thực hiện quản lý các máy ảo chạy tại các nút.

**Networking:** Các nút công khai, nút phụ trợ và các máy ảo được đặt trong một mạng con riêng biệt nhằm đảm bảo tính bảo mật, an toàn của dữ liệu.

**Load balancer:** Lớp cân bằng tải thực hiện phân phối các yêu cầu từ internet tới các nút công cộng. Đối với tính tin cậy và khả năng mở rộng của hệ thống, mỗi dịch vụ được nhân rộng trên nhiều máy ảo. Tuy nhiên, vì các dịch vụ cũng tương đối nhẹ (so với các ứng dụng nguyên khối) nên nhiều dịch vụ thường được đóng gói thành một máy ảo duy nhất.

#### c) Kiến trúc được sử dụng khi nào

Kiến trúc microservice thường được dùng trong một số trường hợp sau:

- Các ứng dụng lớn, yêu cầu tốc độ phát hành cao.
- Các ứng dụng phức tạp, cần có khả năng mở rộng cao.
- Ứng dụng với các miền đa dạng, sử dụng kết hợp nhiều công nghệ.
- Trong một tổ chức có nhiều nhóm phát triển nhỏ.

#### d) Ưu điểm

- **Triển khai độc lập:** Bạn có thể cập nhật một dịch vụ mà không cần thiết phải triển khai lại toàn bộ hệ thống. Dễ dàng hơn trong việc sửa lỗi và phát hành các tính năng mới ít rủi ro hơn.
- **Phát triển độc lập:** Các nhóm phát triển có thể xây dựng, thử nghiệm và triển khai dịch vụ một cách độc lập. Do đó sự đổi mới và khả năng phát hành nhanh hơn.
- **Các nhóm nhỏ, tập trung:** Các nhóm có thể tập trung vào một dịch vụ. Các dịch vụ có phạm vi nhỏ hơn làm cho mã nguồn dễ hiểu hơn, do đó thành viên mới vào có thể dễ dàng hòa nhập hơn.
- **Cô lập lỗi:** Nếu một dịch vụ bị lỗi, nó sẽ không làm hỏng toàn bộ ứng dụng. Việc sửa lỗi sẽ chỉ cần tập trung vào các dịch vụ đang có lỗi. Các dịch vụ khác vẫn có thể hoạt động bình thường.
- **Kết hợp công nghệ:** Các nhóm phát triển có thể lựa chọn các công nghệ phù hợp dựa trên những lợi ích của từng công nghệ để kết hợp lại với nhau.

#### e) Hạn chế

- **Tính phức tạp:** Một ứng dụng microservices có nhiều bộ phận cùng hoạt động hơn so với ứng dụng đơn khối (monolithic). Mỗi dịch vụ đơn giản hơn, nhưng toàn bộ hệ thống thì càng phức tạp



- **Phát triển và kiểm thử:** Việc phát triển dựa trên các dịch vụ phụ thuộc đòi hỏi một cách tiếp cận khác. Các công cụ hiện có có thể không được phát triển để làm việc với các dịch vụ phụ thuộc.
- **Thiếu sự quản lý:** Cách tiếp cận phân cấp để xây dựng các dịch vụ nhỏ có lợi thế, nhưng nó cũng có thể dẫn đến các vấn đề. Các ứng dụng có thể được xây dựng bằng nhiều ngôn ngữ và các framework khác nhau do đó nó làm cho ứng dụng trở nên khó quản lý và bảo trì hơn.
- **Các dịch vụ phải giao tiếp trên mạng,** nên tốc độ có thể không cao bằng ứng dụng monolithic
- **Toàn vẹn dữ liệu.** Với mỗi microservice phải tự chịu trách nhiệm về tính toàn vẹn dữ liệu của riêng mình. Do đó yêu cầu thống nhất dữ liệu có thể trở thành một thách thức lớn.
- **Quản lý phiên bản.** Việc cập nhật cho một dịch vụ phải đảm bảo không phá vỡ các dịch vụ phụ thuộc vào nó. Các dịch vụ có thể được cập nhật ở bất cứ thời điểm nào. Vì vậy nếu không được thiết kế cẩn thận, có thể gặp sự cố với tính tương thích của các dịch vụ.
- **Yêu cầu về kỹ năng:** Microservice là hệ thống có tính phân tán cao, do đó phải đánh giá cẩn thận liệu đội ngũ kỹ thuật đã có đủ kỹ năng và kinh nghiệm để triển khai thành công hay không.

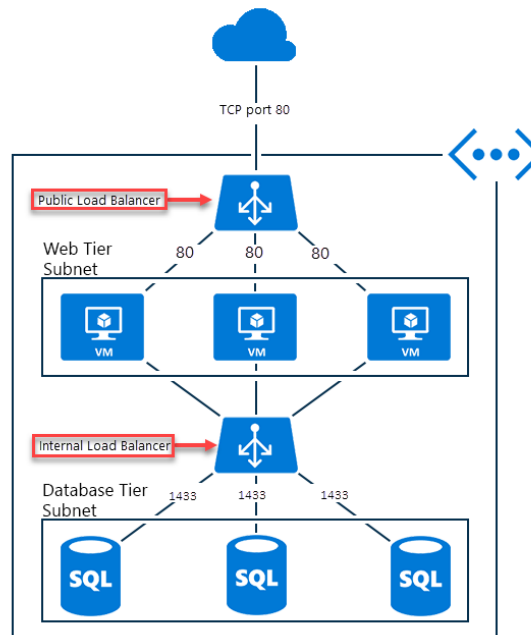
### 2.3. Các yếu tố ảnh hưởng đến khả năng chịu tải của hệ thống

Ba yếu tố quan trọng nhất khi thiết kế một hệ thống có khả năng chịu tải cao đó là: Hiệu năng (Performance), tính sẵn sàng (Availability) và khả năng mở rộng hệ thống (Scalability).

- **Performance:** Thể hiện tốc độ phản hồi của hệ thống, được đo bằng đơn vị thời gian, có thể là giây hoặc mili giây. Hệ thống hoạt động càng nhanh thì người dùng làm được nhiều việc hơn, đem lại lợi nhuận cao hơn. Hệ thống mà quá chậm thì sẽ không có ai sử dụng.
- **Availability:** Chỉ khả năng hoạt động của hệ thống vào mọi thời điểm, được đo bằng uptime. Ví dụ trong 1 ngày, nếu hệ thống hoạt động 100 ngày và 1 ngày gặp sự cố không thể chạy được thì  $\text{uptime} = 99/100 = 99\%$ .
- **Scalability:** Khả năng mở rộng của hệ thống. Liệu khi có đông user hơn thì hệ thống có thể mở rộng (scale) được không? Việc scale có thể thực hiện dễ dàng, nhanh chóng hay không?.

### 2.3.1. Đảm bảo hiệu năng (Performance)

- **Cân bằng tải với Load balancer:** Load Balancer là một thiết bị (phần cứng hoặc phần mềm) cho phép cân bằng tải đến nhiều server. Giả sử ta có 1 server có thể phục vụ 1000 người. Để phục vụ 10000 người, ta có thể chạy 10 server. Người dùng sẽ không trực tiếp truy cập tới server, mà chỉ truy cập tới load balancer. Bộ cân bằng tải sẽ điều tiết, cân bằng lượng tải trên 10 server này.



Hình 2.17: Cân bằng tải với Load Balancer

Azure load balancer cung cấp 2 khái niệm về load balancer là public load balancer và internal load balancer:

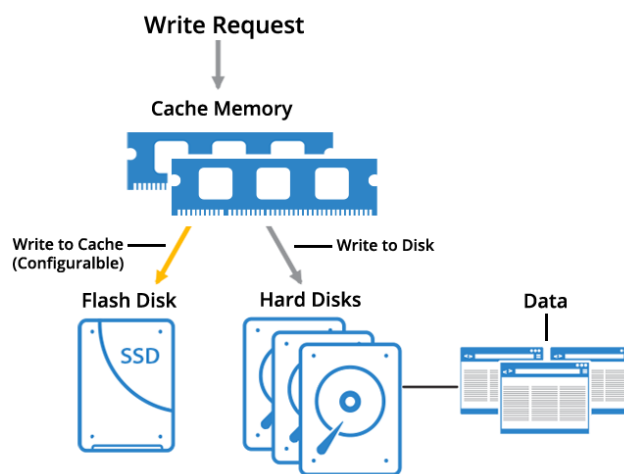
- Public load balancer: là nơi tiếp nhận và điều phối các request của người dùng tới các web server tương ứng
  - Internal load balancer: là nơi điều phối lưu lượng tới các tài nguyên bên trong hệ thống
- **Phân tán dữ liệu với Content Delivery Network (CDN):** dịch vụ lưu trữ các nội dung tĩnh của website: html, javascript, css, image,... và đặt các nội dung này ở các máy chủ khác nhau, giúp tối ưu hóa việc truy cập của user ở nhiều nơi trên thế giới.



Hình 2.18: Cách thức hoạt động của CDN

Những lợi ích khi sử dụng CDN để phân phối tài nguyên của website bao gồm:

- Tiết kiệm băng thông đáng kể đối với các dữ liệu tĩnh (hình ảnh, css, javascript)
  - Tăng tốc độ truy cập website, load nội dung nhanh, giảm thiểu độ trễ, giật hình khi truy cập và xem các trang website phân phối nội dung như: Movies, Video clip, vvv...
  - Cho phép người dùng xem các chương trình, sự kiện truyền hình trực tuyến trên Internet thông qua máy tính, laptop, các thiết bị cầm tay với tốc độ nhanh nhất, đảm bảo chất lượng hình ảnh, âm thanh tốt nhất mà không cần phải đầu tư hay trang bị các thiết bị truyền hình đắt tiền nào khác
- **Caching:** Azure redis cache là giải pháp cache dữ liệu trong bộ nhớ, giúp giảm số lượng request tới cơ sở dữ liệu. Tất cả mọi request đến đều phải lấy dữ liệu từ cache trước tiên, trường hợp trong cache chưa có dữ liệu mới thực hiện lấy từ database.

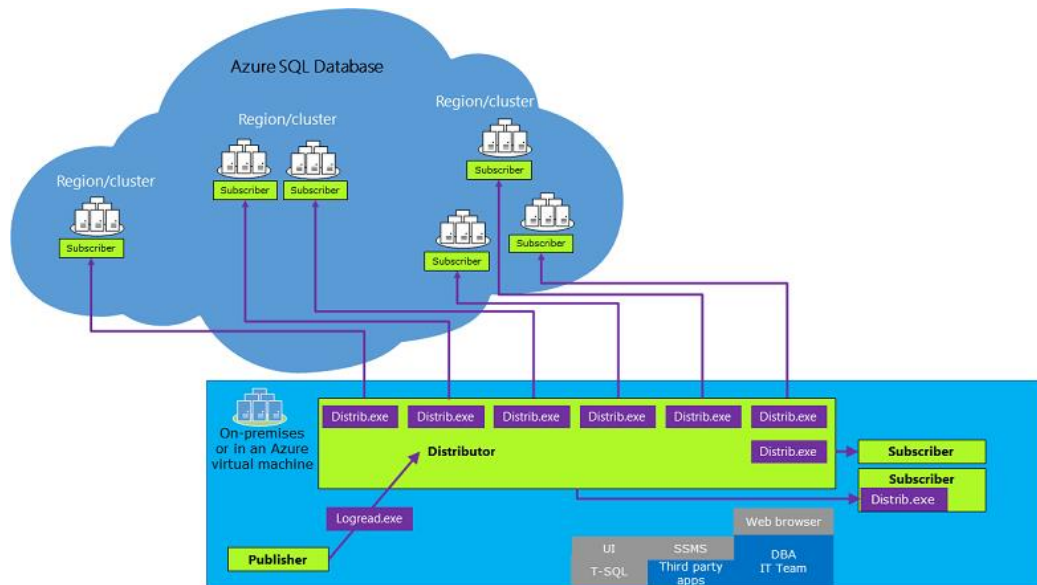


Hình 2.19: Cách thức hoạt động của Caching

### 2.3.2. Đảm bảo tính sẵn sàng của hệ thống (Availability)

**Master/Slave:** Thay vì chỉ chạy 1 server, ta chạy 2 hoặc nhiều hơn. 1 server chính gọi là master, các server còn lại là slave. Khi master có vấn đề (sập nguồn hay crash), một slave sẽ được chỉ định để lên thay thế master.

**Replication:** Thường được kết hợp chung với Load Balancer. Code của ứng dụng sẽ được deploy lên nhiều server. Khi có 1 server không hoạt động, load balancer sẽ chuyển request sang server khác, đảm bảo request vẫn được thực hiện.



Hình 2.20: Kiến trúc Master/ Slave trong Azure SQL

### 2.3.3. Đảm bảo tính mở rộng hệ thống (Scalability)

Khả năng mở rộng của ứng dụng là giá trị đo số lượng người dùng mà ứng dụng có thể phục vụ trong cùng một lúc. Điểm mà tại đó ứng dụng không thể xử lý thêm được người dùng chính là giới hạn khả năng mở rộng của nó. Khả năng mở rộng đạt đến giới hạn của nó khi một tài nguyên phần cứng quan trọng hết. Có hai cách để tăng khả năng đáp ứng của hệ thống:

#### a) Mở rộng theo chiều dọc (Vertical scaling):

Là việc tăng sức mạnh phần cứng cho server bằng cách... lắp thêm RAM, lắp thêm CPU, thay ổ cứng bằng ổ SSD, tăng băng thông mạng. Cách này khá nhanh và dễ thực hiện. Nhược điểm của cách làm này là: phần cứng luôn có một giới hạn, phần cứng

Việc mở rộng cấu hình trên Azure rất dễ dàng, chỉ cần một vài thao tác trên portal là có thể tăng được sức mạnh phần cứng cho ứng dụng. Azure cung cấp các dịch vụ với các thông tin cấu hình phần cứng khác nhau.

STT	Series	Instance	Cores	Ram	Storage	Pricing
1	Basic Service Plan	B1	1	1.75 GB	10 GB	\$0.075/hour
2		B2	2	3.50 GB	10 GB	\$0.15/hour
3		B3	4	7 GB	10 GB	\$0.30/hour
4	Av2 Standard	S1	1	1.75 GB	50 GB	\$0.10/hour
5		S2	2	3.50 GB	50 GB	\$0.20/hour
6		S3	4	7 GB	50 GB	\$0.40/hour
7	Premium Service Plan	P1v2	1	3.50 GB	250 GB	\$0.20/hour
8		P2v2	2	7 GB	250 GB	\$0.40/hour
9		P3v2	4	14 GB	250 GB	\$0.80/hour
10	Isolated Service Plan	I1	1	3.50 GB	1 TB	\$0.40/hour
11		I2	2	7 GB	1 TB	\$0.80/hour
12		I3	4	14 GB	1 TB	\$1.60/hour

Bảng 2.1: Bảng thông số cấu hình của dịch vụ Azure Web app

**b) Mở rộng theo chiều ngang (Horizontal scaling):**

Thay vì tăng sức mạnh cho 1 server, ta thêm nhiều server vào hệ thống và chạy cùng lúc. Cách này làm cho thiết kế phức tạp hơn, nhưng bù lại ta có thể dùng phần cứng rẻ tiền lắp thêm vào, khả năng mở rộng hầu như không giới hạn.

STT	Serie	Maximum instances	Auto Scale	Custom domain
1	Basic Service Plan	3	Không hỗ trợ	Hỗ trợ
4	Av2 Standard	10	Hỗ trợ	Hỗ trợ
7	Premium Service Plan	20	Hỗ trợ	Hỗ trợ
10	Isolated Service Plan	100	Hỗ trợ	Hỗ trợ

Bảng 2.2: Bảng thông số instance tối đa có thể mở rộng của Azure Web app

Chúng ta có thể cấu hình auto scale theo lịch hoặc theo mức độ sử dụng tài nguyên hệ thống như % CPU, % RAM,...

Microsoft Azure cung cấp cơ chế tự động scale cho các ứng dụng web với các thành phần:

- Web app, API app cung cấp cơ chế tự động scale hệ thống bằng cách tăng hoặc giảm số lượng web instance. Người dùng có thể cấu hình tự động scale hệ thống một cách dễ dàng qua azure portal.
- Có thể cấu hình auto scale theo thời gian:
- Hoặc có thể cấu hình tự động mở rộng hệ thống khi tài nguyên của hệ thống được sử dụng đạt đến một ngưỡng nhất định:

## 2.4. Kết luận

Chương này cung cấp một cái nhìn tổng quan về các thành phần dịch được cung cấp trên nền tảng đám mây Microsoft Azure. Chúng ta có thể thấy được rằng Azure cung cấp một kho dịch vụ khổng lồ từ các dịch vụ cơ bản về hạ tầng là các máy ảo đến các dịch vụ nền tảng lưu trữ và phân phối dịch vụ đa phương tiện như Media Service. Dựa trên các dịch vụ được cung cấp sẵn này, các lập trình viên và nhà phát triển phần mềm có thể thỏa sức sáng tạo dựa trên một hạ tầng có thể nói là không có giới hạn.

Chương 2 cũng đưa ra 3 kiểu kiến trúc phần mềm ứng dụng các dịch vụ trên Microsoft Azure để tận dụng sức mạnh của nền tảng này, đó là các kiểu kiến trúc: Kiến trúc phân tầng, kiến trúc Web – Queue – Worker, kiến trúc vi dịch vụ. Nhờ các tính năng hỗ trợ hệ thống replicate ở các vùng khác nhau và khả năng tự động mở rộng tài nguyên hệ thống của Azure mà chúng ta có thể xây dựng được những hệ thống có khả năng đáp ứng một số lượng lớn người dùng đồng thời cùng một lúc.

Nền tảng Azure đang thu hút được khá nhiều nhà phát triển ứng dụng với các hệ thống có quy mô lớn, cần tới khả năng mở rộng cả về năng lực xử lý và khả năng lưu trữ. Tuy nhiên trở ngại lớn nhất của Microsoft Azure hiện tại chính là nằm ở vấn đề về mức giá còn tương đối cao.

## **Chương 3. MỘT MÔ HÌNH ỨNG DỤNG KIẾN TRÚC PHẦN MỀM TRÊN NỀN TẢNG CÔNG NGHỆ AZURE CỦA MICROSOFT**

Trên cơ sở lý thuyết về các kiểu kiến trúc phần mềm trên nền tảng điện toán đám mây và các dịch vụ được cung cấp trên Azure của Microsoft ở chương 2, trong chương này tôi xây dựng kiến trúc phần mềm cho ứng dụng chấm công bằng khuôn mặt.

### **3.1. Mô tả bài toán**

#### **3.1.1. Giới thiệu**

Hiện nay, việc thực hiện chấm công thời gian làm việc của nhân viên tại các công ty, tập đoàn lớn thường sử dụng các giải pháp chấm công bằng vân tay, chấm công bằng khuôn mặt. Tuy nhiên các giải pháp này có những nhược điểm như:

- Thời gian thực hiện chậm trễ
- Nhân viên phải xếp hàng chờ đợi đến lượt chấm công
- Có thể nhờ chấm hộ khi sử dụng thẻ của nhau

#### **3.1.2. Giải pháp**

Chương trình chấm công bằng khuôn mặt này sẽ áp dụng những công nghệ được cung cấp bởi Azure để thấy được những lợi ích sử dụng của công nghệ này. Các dịch vụ được cung cấp bởi Azure.

Chương trình chấm công bằng khuôn mặt sử dụng dịch vụ xác thực khuôn mặt Face ID APIs của Microsoft để thực hiện nhận dạng với cơ sở dữ liệu nhân sự đã được đăng ký trước đó để xác định lượt ra vào. Thời gian chấm công sẽ được lấy bằng thời gian đầu tiên và cuối cùng của nhân viên được nhận diện.

### **3.2. Phân tích nghiệp vụ**

#### **3.2.1. Mô tả chức năng**

##### **a) Quản lý hồ sơ nhân viên**

- Nhân viên khi vào làm việc, phải nộp hồ sơ ban đầu bao gồm: Đơn xin việc, sơ yếu lý lịch, giấy khám sức khỏe, bằng cấp chuyên môn.
- Thông tin nhân viên cần cập nhật bao gồm: Mã nhân viên, mã phòng ban, họ tên nhân viên, giới tính, ngày sinh, địa chỉ thường trú, địa chỉ hiện tại, số chứng minh nhân dân, số bảo hiểm, quê quán, dân tộc, tôn giáo, bằng cấp, quá trình công tác,...

- Training dữ liệu ban đầu: mỗi nhân viên sẽ được chụp 5 ảnh ban đầu (bao gồm các góc độ chụp ảnh khác nhau) để thực hiện làm bộ dữ liệu training cho quá trình nhận diện hình ảnh về sau.

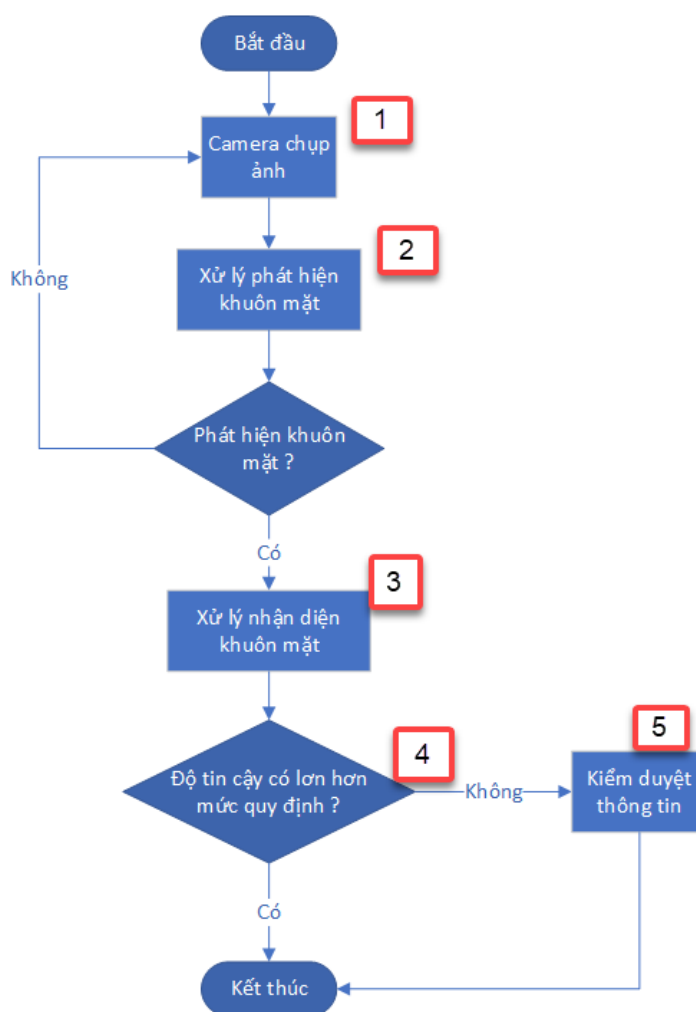
### b) Quản lý chấm công

- Việc chấm công được thực hiện tự động khi nhân viên xuất hiện trước camera. Thông tin chấm công hàng ngày được cập nhật vào cơ sở dữ liệu của chương trình.
- Bảng chấm công bao gồm: số thứ tự, họ tên nhân viên, số ngày làm việc, số ngày nghỉ phép, số ngày nghỉ phép, số ngày nghỉ có lương,...

### 3.2.2. Quy trình chấm công bằng khuôn mặt

#### a) Mô hình quy trình xử lý

Việc chấm công được thực hiện tự động khi nhân viên xuất hiện trước camera. Thông tin chấm công hàng ngày được cập nhật vào cơ



Hình 3.1: Quy trình chấm công bằng khuôn mặt



## **b) Mô tả các bước trong quy trình**

### **• Camera chụp ảnh (bước 1)**

- Đối tượng thực hiện: Hệ thống tự động xử lý
- Đầu vào: N/A
- Đầu ra: Ảnh chụp
- Mô tả: Hệ thống tự động chụp ảnh từ Camera với tần số lấy mẫu là 2 lần trên 1 giây (thông số này có thể thay đổi theo thông tin cấu hình).

Khi hệ thống chụp được 1 khung hình, sẽ tiến hành bước xử lý phân tích hình ảnh để tìm kiếm khuôn mặt trên khung hình

### **• Xử lý phát hiện khuôn mặt (bước 2)**

- Đối tượng thực hiện: Hệ thống tự động xử lý
- Đầu vào: Ảnh chụp
- Đầu ra: Danh sách khuôn mặt phát hiện được
- Mô tả: Sử dụng các thuật toán và thư viện xử lý hình ảnh để phát hiện khuôn mặt.

Sau khi xử lý hình ảnh, nếu phát hiện được khuôn mặt thì sẽ chuyển đến bước xử lý nhận diện khuôn mặt, còn trong trường hợp không phát hiện được khuôn mặt thì sẽ tiến hành quay lại bước chụp ảnh.

### **• Xử lý nhận diện khuôn mặt (bước 3)**

- Đối tượng thực hiện: Hệ thống tự động xử lý
- Đầu vào: Danh sách khuôn mặt đã phát hiện
- Đầu ra: Thông tin khuôn mặt giống nhất, độ tin cậy
- Mô tả: Thực hiện kết nối với dịch vụ FaceID của Microsoft để tiến hành nhận dạng.

### **• Đánh giá mức độ tin cậy (bước 4)**

- Đối tượng thực hiện: Hệ thống tự động xử lý
- Đầu vào: Thông tin khuôn mặt giống nhất, độ tin cậy
- Đầu ra: Dữ liệu chấm công
- Mô tả: Hệ thống sẽ tự động đánh giá kết quả nhận dạng theo ngưỡng độ tin cậy cho phép

Khi độ tin cậy trả về lớn hơn ngưỡng đánh giá, thì hệ thống tự động lưu trữ dữ liệu chấm công của nhân viên. Còn trường hợp độ tin cậy thấp thì sẽ phải thực hiện bước kiểm duyệt lại thông tin hình ảnh chấm công.

### **• Kiểm duyệt thông tin (bước 5)**

- Đối tượng thực hiện: Nhân viên quản lý chấm công
- Đầu vào: Hình ảnh chấm công
- Đầu ra: Thông tin chấm công
- Mô tả: Nhân viên quản lý chấm công thực hiện xem lại hình ảnh ra vào của nhân viên trong những trường hợp có độ tin cậy thấp. Trường hợp là nhân viên trong công ty thì chuyển thành dữ liệu chấm công, và có thể cập nhật lại bộ training.

### 3.2.3. Biểu đồ các trường hợp sử dụng (Use Case)

Tác động vào hệ thống có 2 nhóm tác nhân chính là: Quản trị hệ thống, Quản lý chấm công. Người dùng thuộc nhóm Quản trị hệ thống có quyền thực hiện tất cả chức năng trong hệ thống.

Trường hợp sử dụng tổng quan:

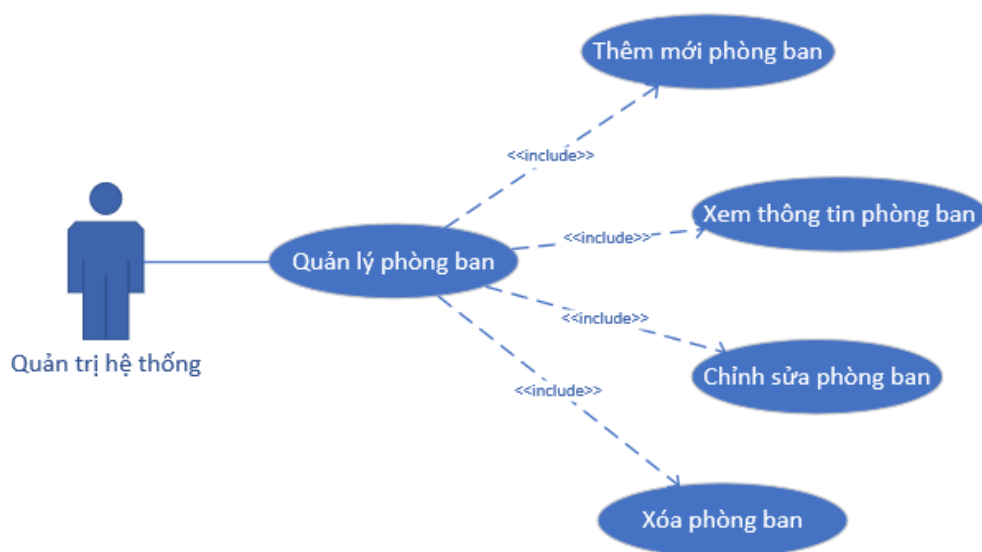


Hình 3.2: Các trường hợp sử dụng tổng quan

Trong trường hợp tổng quan, người quản trị hệ thống có thể thực hiện các chức năng: Quản lý thông tin phòng ban, Quản lý thông tin nhân viên, Quản lý lịch làm việc, Quản lý thời gian chấm công, Quản lý người dùng trong hệ thống, Quản lý các thông tin cấu hình, Thống kê, báo cáo.

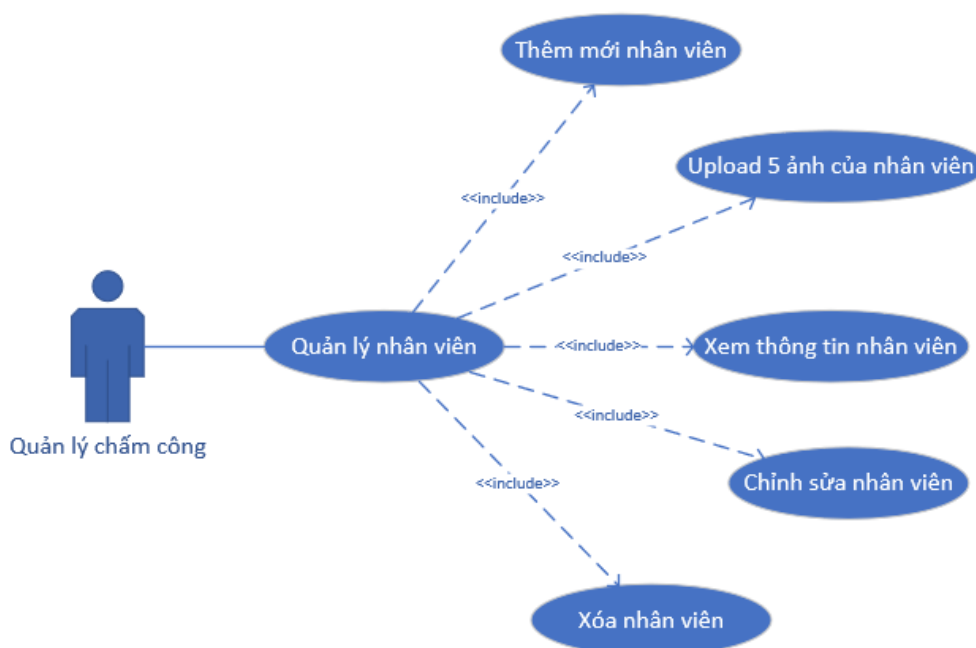
Một số trường hợp sử dụng cụ thể:

- Quản lý phòng ban:



Hình 3.3: Use case Quản lý phòng ban

- Quản lý nhân viên:

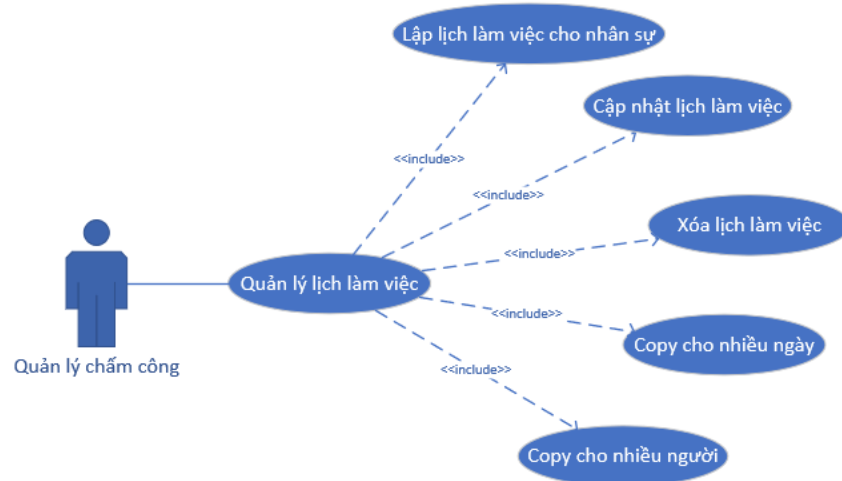


Hình 3.4: Use case Quản lý nhân viên

Người dùng thuộc nhóm Quản lý chấm công có thể thực hiện các chức năng:

- Thêm mới thông tin của nhân viên: Mã nhân viên, tiền nhân viên, tên phòng ban, chức vụ,...

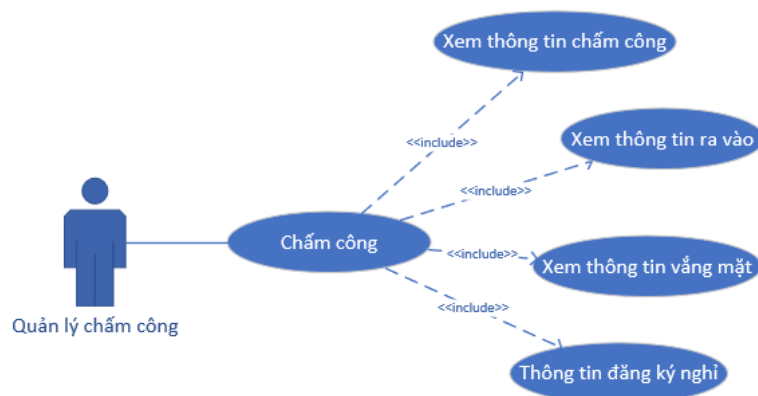
- Upload 5 ảnh cận mặt của nhân viên với các góc chụp khác nhau, dùng làm dữ liệu nhận diện khuôn mặt của nhân viên.
- Tra cứu, xem thông tin của nhân viên
- Chính sửa thông tin của nhân viên
- Xóa nhân viên
- Quản lý lịch làm việc:



Hình 3.5: Use case Quản lý lịch làm việc

Use case này thực hiện chức năng hiển thị hoặc cập nhật lịch làm việc chi tiết đến từng ngày cho nhân sự theo các ca làm việc và xem thông tin đăng ký nghỉ có lý do.

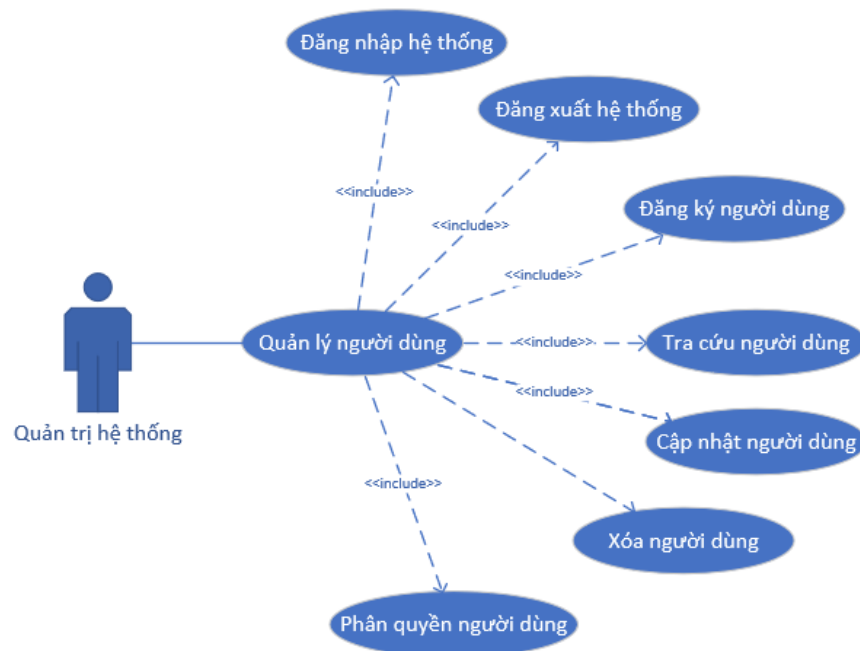
- Chấm công:



Hình 3.6: Use case Chấm công

Trường hợp này, người quản lý chấm công xem các thông tin chấm công, thông tin ra vào của từng nhân viên theo một khoảng thời gian, theo phòng ban,....

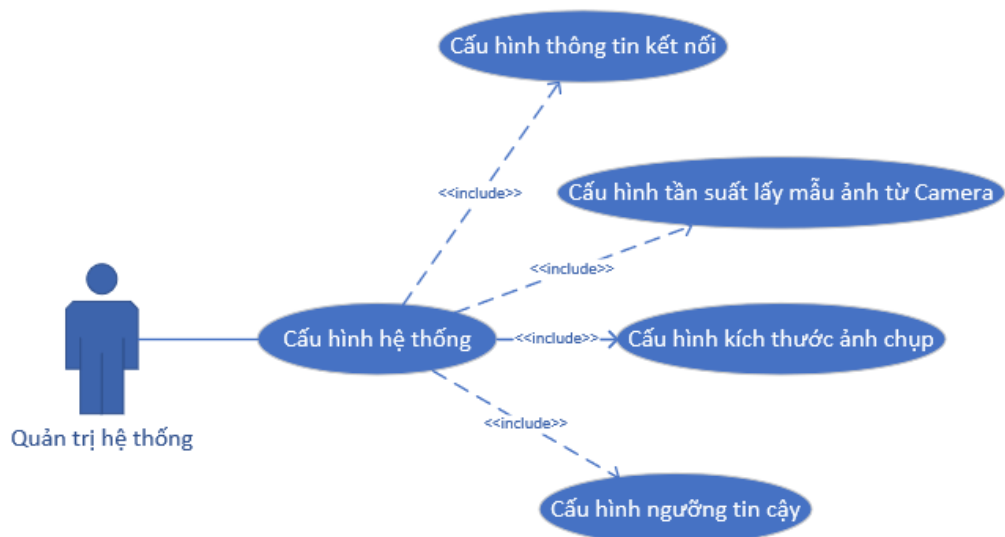
- Quản lý người dùng:



Hình 3.7: Use case Quản lý người dùng

Người dùng thuộc nhóm Quản trị hệ thống có thể tạo mới người dùng trong hệ thống, cấp quyền cho người dùng vào các chức năng trong hệ thống. Đồng thời người quản trị cũng có thể chỉnh sửa thông tin người dùng cũng như xóa người dùng trong trường hợp không sử dụng đến.

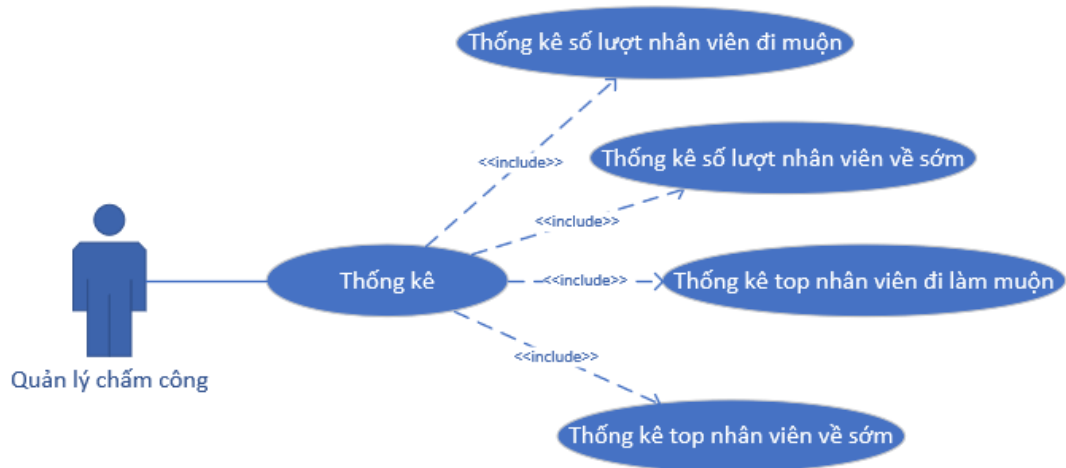
- Cấu hình hệ thống:



Hình 3.8: Use case Cấu hình hệ thống

Người quản trị hệ thống có thể sử dụng các tính năng: cấu hình các thông tin kết nối tới Service Bus, Azure Storage, Cấu hình tần suất lấy mẫu ảnh từ Camera, Cấu hình kích thước ảnh chụp từ Camera, Cấu hình ngưỡng tin cậy khi xác thực hình ảnh nhân viên.

- Thống kê:



Hình 3.9: Use case Thống kê

Người dùng thuộc nhóm Quản lý chấm công có thể xem được tổng hợp số lượt nhân viên đi làm sớm, về muộn theo từng ngày, từng tuần, theo khoảng thời gian. Tổng hợp số lượng nhân viên đi sớm, về muộn theo tháng, theo quý.

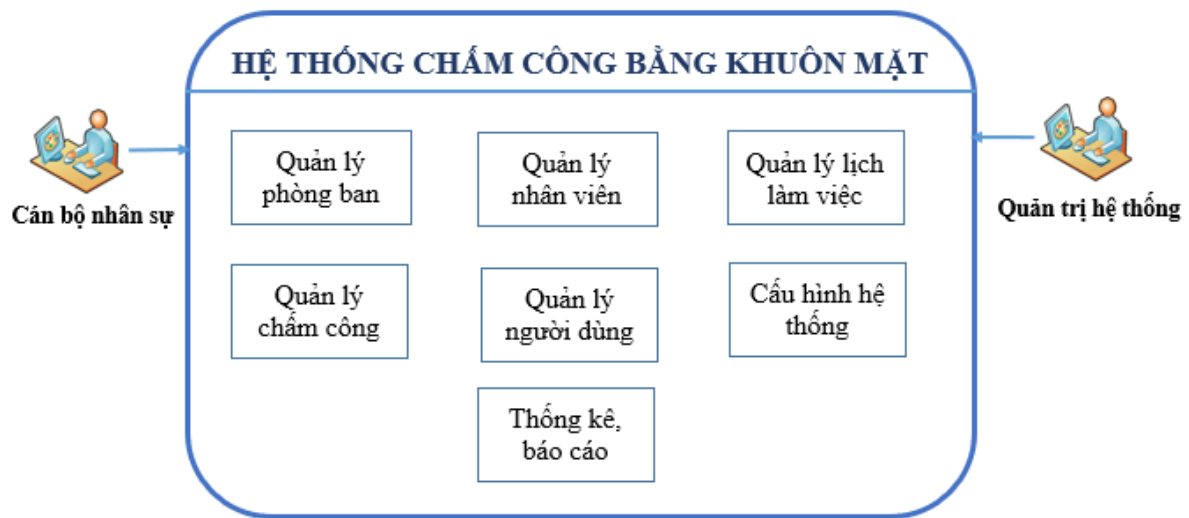
#### 3.2.4. Các module chức năng hệ thống

Hệ thống cần phải có các khối chức năng chính sau đây:

- Quản lý phòng ban
- Quản lý nhân viên
- Nhận diện khuôn mặt
- Quản lý chấm công
- Quản lý lịch làm việc
- Quản trị hệ thống: cấu hình hệ thống, quản lý người dùng, phân quyền hệ thống.
- Thống kê, báo cáo

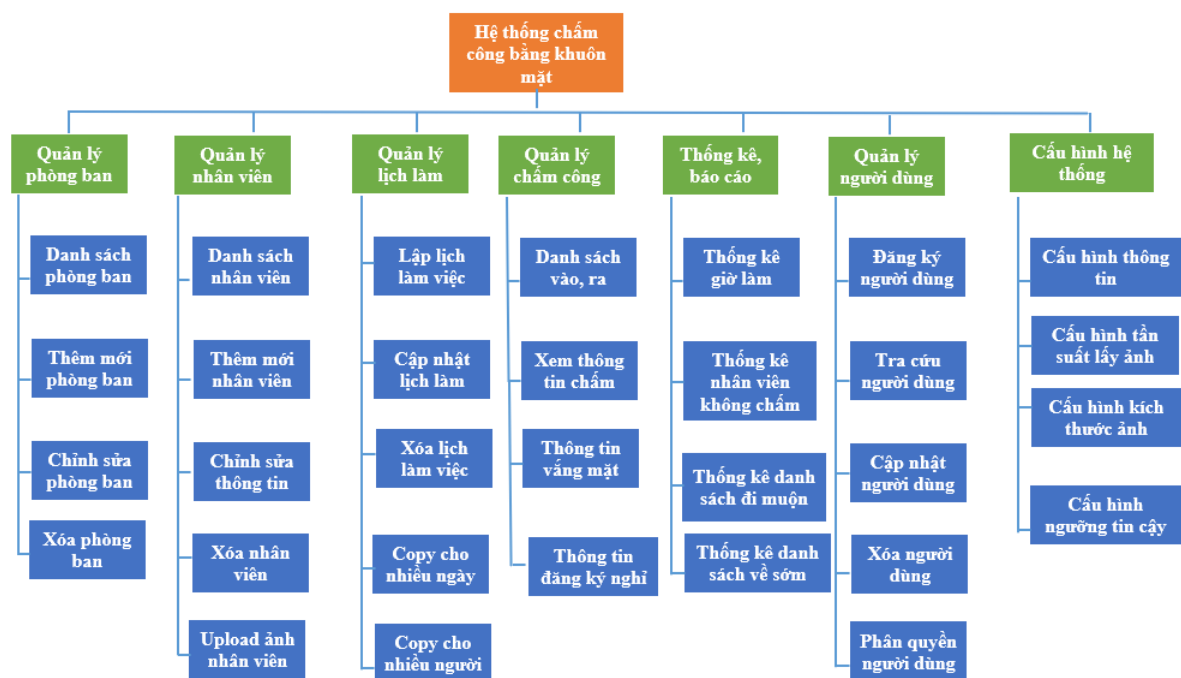
### 3.3. Thiết kế hệ thống

#### 3.3.1. Mô hình tổng thể chức năng hệ thống



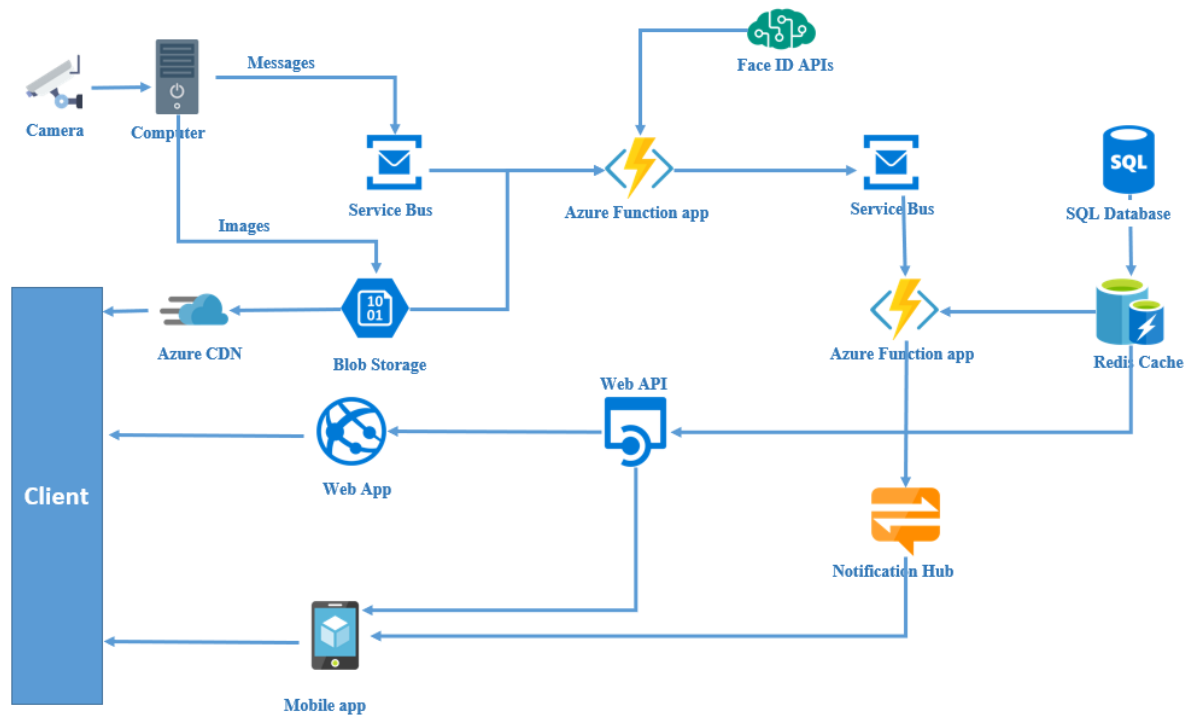
Hình 3.10: Mô hình tổng thể chức năng hệ thống

#### 3.3.2. Mô hình phân rã chức năng



Hình 3.11: Mô hình phân rã chức năng

### 3.3.3. Kiến trúc hệ thống



Hình 3.12: Kiến trúc hệ thống

Các thành phần dịch vụ Azure được sử dụng trong hệ thống:

- **Service bus:** Máy tính liên tục chụp ảnh từ camera và tiến hành xử lý hình ảnh. Khi phát hiện khuôn mặt trên ảnh, máy tính sẽ gửi 1 bản tin lên queue FaceRecognition trên Service bus. Đồng thời máy tính cũng thực hiện lưu 1 ảnh có chứa khuôn mặt lên Blob.
- **Azure functions app:** được kích hoạt theo trigger của service bus, mỗi khi có 1 bản tin được lưu vào queue. Function app sẽ gọi đến dịch vụ nhận diện và xác thực khuôn mặt Face ID để xác định xem ảnh được nhận diện là của ai.
  - Azure Functions là dịch vụ cho chạy các đoạn code có độ phức tạp của nghiệp vụ không cao trên cloud mà không muốn bận tâm tới các thành phần khác liên quan trong ứng dụng cũng như hạ tầng để chạy được function này.
  - Hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C#, F#, Node.js, PHP hay Python
  - Có khả năng nhanh chóng tích hợp với các dịch vụ khác của Azure như Azure Storage, Azure Mobile Apps, Azure Event Hubs, ...



- Với Azure Functions thay vì phải trả một con số cố định và không hề rẻ để thuê server để chạy function của bạn thì với gói Consumption của Azure Functions, bạn chỉ cần phải trả cho thời gian mà function của bạn chạy mà thôi.
- **Face ID APIs:** dịch vụ nhận diện và xác thực khuôn mặt của Microsoft
  - FaceID là 1 dịch vụ thuộc Microsoft Cognitive Services, trong đó bao gồm một bộ các API ứng dụng trí tuệ nhân tạo thông minh, cho phép lập trình viên phát triển các ứng dụng phát hiện, nhận diện khuôn mặt một cách nhanh chóng.
  - Ngoài việc phát hiện khuôn mặt trong ảnh thì API này cũng trả về các thuộc tính của khuôn mặt như tuổi, giới tính, trạng thái cảm xúc,... Ngoài ra, Face API còn cho phép so sánh 2 khuôn mặt có phải là của cùng một người hay không.
  - Face ID có thể phát hiện tối đa 64 khuôn mặt trên 1 bức hình từ 1 file ảnh được upload hoặc từ 1 file đã có trên web
  - Kích thước file ảnh có thể phát hiện khuôn mặt từ 1Kb đến 4Mb
- **Blob storage:** Sử dụng để lưu trữ hình ảnh training ban đầu của nhân viên và hình ảnh nhận được từ các camera.
  - Blob cung cấp dịch vụ lưu trữ dữ liệu không cấu trúc trên Azure, có thể là file ảnh, file video, file văn bản,...
  - Block blob: dùng để lưu trữ text hoặc file có sẵn. 1 single block blob có thể chứa đến 50000 blocks và mỗi block có thể lên đến 100Mb, nên tổng dung lượng của 1 single block blob có thể lên đến 4.75 TB
  - Append blob: cũng giống như block blob nhưng được tối ưu hóa cho việc thêm dữ liệu vào blob liên tục, nên nó thường được sử dụng để lưu các file logging. Dung lượng của 1 single append blob có thể lên đến 195GB (4Mb x 50000)
  - Page blob: dung lượng có thể lên đến 1 TB
- **Azure SQL:** Lưu trữ cơ sở dữ liệu của hệ thống.
  - Azure SQL được xây dựng trên phần mềm quản lý cơ sở dữ liệu Microsoft SQL Server.
  - Azure SQL được host trên các trung tâm dữ liệu của Microsoft và được lưu trữ phân tán trên rất nhiều nodes. Azure SQL sẽ lưu trữ cơ sở dữ liệu thành nhiều bản copy, do đó nếu có vấn đề gì thì có thể khôi phục lại dữ liệu một cách đơn giản

- Azure SQL giúp bảo vệ dữ liệu bằng cách bắt buộc các kết nối tới đều phải thực hiện mã hóa (SSL/TLS)
- **Redis Cache:** Lưu trữ dữ liệu trong bộ nhớ đệm nhằm cải thiện hiệu năng của hệ thống
- **Notification Hubs:** Mỗi khi nhận diện được hình ảnh của nhân viên, một notify sẽ được gửi tới ứng dụng mobile thông qua Notification Hubs.
- **Web app service:** Triển khai trang frontend của hệ thống
  - Là dịch vụ trên Microsoft Azure được xây dựng để tối ưu cho việc hosting website. App service hỗ trợ rất nhiều ngôn ngữ lập trình như Java, PHP, Node.js, Python và .Net.
  - Azure app service hỗ trợ việc tự động scale hệ thống và tự động load balance bằng cách tăng giảm số lượng các instances một cách tự động khi số lượng request đến hệ thống tăng hoặc giảm.
  - Azure app service cung cấp tính năng giám sát hệ thống, người quản trị có thể xem các thống kê về hiệu năng của hệ thống theo thời gian thực.
- **API app service:** Triển khai trang backend của hệ thống. Chứa các API làm việc với frontend và ứng dụng Mobile
- **Azure CDN:** Lưu trữ các file tĩnh như: Javascript, Css, html, ảnh.

#### 3.3.4. Quy trình xử lý dữ liệu ảnh khi nhận diện



Hình 3.13: Quy trình xử lý ảnh khi nhận diện

Quy trình thực hiện xử lý ảnh nhận diện nhân viên khi xuất hiện trước camera được thực hiện theo các bước sau:

- **Camera chụp ảnh:** Sử dụng giao thức RTSP để kết nối tới luồng stream từ Camera, định kỳ thực hiện capture ảnh từ camera theo tần suất 6 frames/ 1 giây.
- **Xử lý phát hiện khuôn mặt:** Sử dụng thư viện Windows.Media.FaceAnalysis được cung cấp sẵn của nền tảng Windows Universal App để phát hiện khuôn mặt trên ảnh đã chụp.
- **Loại bỏ ảnh có vị trí không thay đổi:** Nếu vị trí của khuôn mặt trên ảnh không có sự thay đổi so với ảnh trước đó thì có thể phán đoán rằng vẫn là 1 người và không cần nhận diện lại nữa.
- **Crop vùng ảnh chứa khuôn mặt:** Chỉ lấy các vùng chứa hình ảnh khuôn mặt trên ảnh.
- **Loại bỏ những khuôn mặt có kích thước nhỏ:** Loại bỏ những ảnh có kích thước quá nhỏ, không có khả năng nhận diện được.
- **Giảm kích thước file ảnh:** Giảm kích thước file ảnh chứa khuôn mặt, để tăng tốc độ nhận diện khuôn mặt.
- **Gửi ảnh lên FaceID APIs để xác thực:** Sau khi thực hiện xác thực được khuôn mặt dựa trên ảnh vừa gửi và tập dữ liệu nhận diện của nhân viên đã có, FaceID API sẽ trả về personId và độ tin cậy, dựa trên độ tin cậy có thể quyết định người được nhận diện là ai.

### 3.4. Xây dựng chương trình thử nghiệm

#### 3.4.1. Môi trường cài đặt, triển khai

Ứng dụng chấm công bằng nhận diện khuôn mặt được xây dựng và triển khai trên nền tảng công nghệ Microsoft Azure, cụ thể các công nghệ và môi trường cài đặt như sau:

- Hệ điều hành: Windows Azure
- Hệ quản trị cơ sở dữ liệu: Microsoft Azure SQL
- Công nghệ lập trình ứng dụng web: .Net, AngularJS, HTML5, XML, CSS, Entity Framework
- Nền tảng công nghệ: .NET
- Máy chủ web: IIS
- Công cụ lập trình: Visual Studio 2017, Window Azure SDK for .NET

### 3.4.2. Các bước triển khai ứng dụng

#### a) Cài đặt các dịch vụ trên Azure

- **Thực hiện tạo Service bus với các thông tin sau:**

- Name: TimeAttendance (đặt tên cho dịch vụ, yêu cầu không được trùng nhau)
- Location: Chọn Southeast Asian (lựa chọn nơi đặt máy chủ dịch vụ Service bus của Microsoft)
- Pricing Tier: Standard (với gói dịch vụ này, tối đa có 12.5 triệu xử lý thao tác với service bus. Kích thước tối đa 1 bản tin là 256 KB). Chi phí của gói này là 10\$/ tháng
- Thực hiện tạo 2 Queue lần lượt là:
  - Face-detection: Chứa thông tin để thực hiện xác thực nhân viên và chấm công
    - Kích thước tối đa của Queue: 4GB
    - Thời gian tồn tại của message là 14 ngày
    - Thời gian khóa message là 30 giây
  - Notification: Chứa thông tin thông báo chấm công thành công để báo về ứng dụng trên mobile.
    - Kích thước tối đa của Queue: 1GB
    - Thời gian tồn tại của message là 1 ngày
    - Thời gian khóa message là 30 giây

- **Cấu hình dịch vụ Azure functions app:**

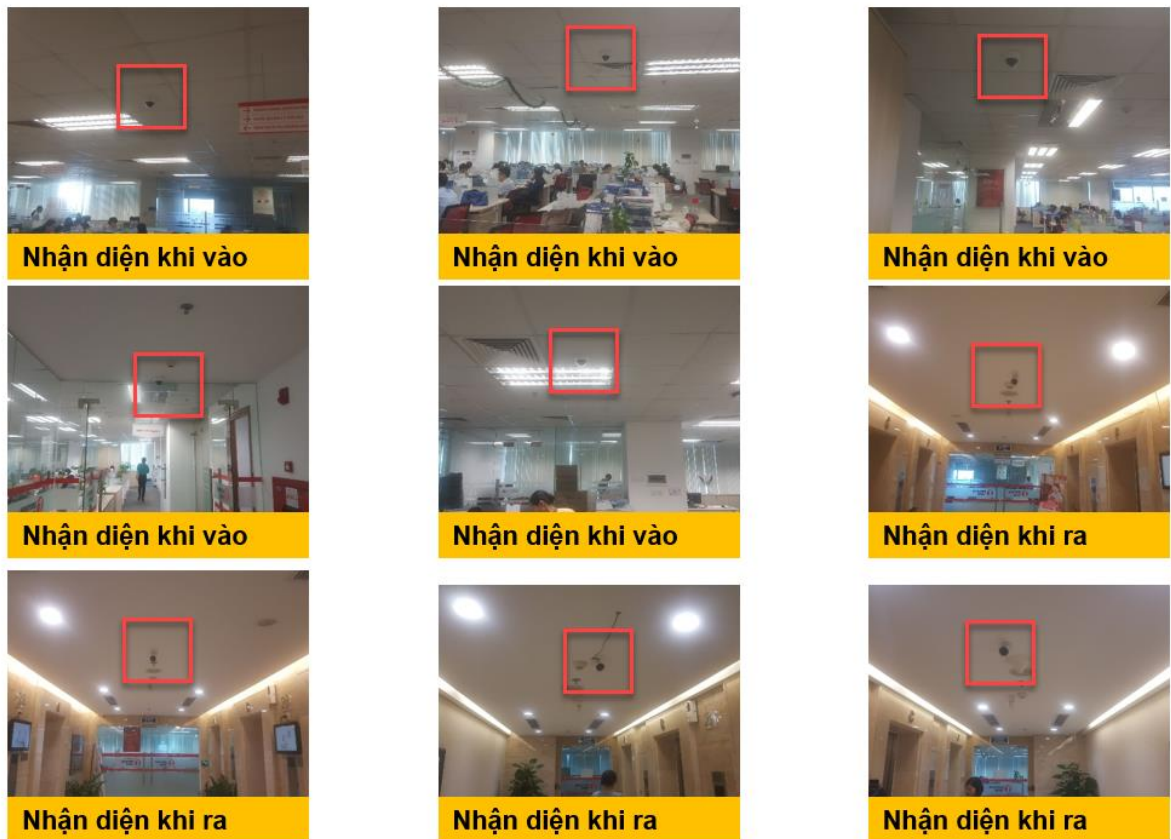
- Name: time-attendance-func (đặt tên cho dịch vụ, yêu cầu không được trùng nhau)
- Location: Southeast Asian
- OS: Windows
- Hosting Plan: Consumption Plan (chi phí được tính theo số lần thực hiện xử lý, \$0.20 cho 1 triệu lượt xử lý)
- Runtime Stack: .NET (cho ứng dụng chạy trên nền tảng .Net)
- Application Insights: Off
- Thực hiện tạo 2 Function lần lượt là:
  - FaceRecognition: chức năng được kích hoạt khi có sự kiện enqueue vào queue có tên là Face-detection

- Notification: chức năng được kích hoạt khi có sự kiện enqueue vào queue có tên là Face-detection
- **Cấu hình dịch vụ Face ID:** cấu hình dịch vụ Cognitive với thông tin sau:
  - Name: TimeAttendance (tên là duy nhất)
  - Location: lựa chọn Southeast Asian (lựa chọn nơi đặt máy chủ dịch vụ Face ID của Microsoft)
  - Pricing Tier: Lựa chọn S0 (FaceID cung cấp 2 gói dịch vụ, F0: gói miễn phí, người dùng được thực hiện 20 transaction trong 1 phút, và tối đa 30.000 transaction trong 1 tháng, và gói S0: người dùng được phép thực hiện tối đa 10 transaction trong 1 giây). Chi phí là 1\$ cho 1.000 transactions
- **Cấu hình dịch vụ Blob storage:**
  - Storage account name: timeattendance (Tên chỉ được chứa các ký tự chữ và số, độ dài từ 3 đến 24 ký tự và không được trùng nhau)
  - Location: Southeast Asian
  - Performance: Standard
  - Account kind: BlobStorage (do chỉ sử dụng để lưu các file ảnh)
  - Tạo container: time-attendance-container
  - URL sau khi tạo: <https://timeattendance.blob.core.windows.net/>
- **Cấu hình dịch vụ Azure SQL với thông tin sau:**
  - Database name: TimeAttendance (tên là duy nhất)
  - Select source: Blank Database
  - Server: TimeAttendance
  - Location: lựa chọn Southeast Asian
  - Pricing Tier: Standard S0 (10 DTU, 250 GB)
- **Cấu hình dịch vụ Redis Cache với thông tin sau:**
  - DNS name: timeattendance (tên là duy nhất)
  - Location: lựa chọn Southeast Asian
  - Pricing Tier: Standard C1 (1 Gb Cache, Replication)
  - URL sau khi tạo: [timeattendance.redis.cache.windows.net:6380](https://timeattendance.redis.cache.windows.net:6380)
- **Tạo dịch vụ Notification Hubs với thông tin sau:**
  - Notification Hub: TimeAttendance (tên là duy nhất)
  - Create new Namespace:
  - Location: lựa chọn Southeast Asian

- Pricing Tier: B1 Basic (10 triệu push và 200.000 thiết bị), chi phí 10\$/tháng
- **Web app service:**
  - App name: time-attendance (tên là duy nhất)
  - OS: Windows
  - Publish: Code
  - Location: Southeast Asian
  - Pricing Tier: Standard S1 (CPU: 1 Core, Ram: 1.75 GB), chi phí 74.4\$/tháng
  - URL sau khi tạo: <https://time-attendance.azurewebsites.net>
- **API app service:** Triển khai trang backend của hệ thống. Chứa các API làm
  - App name: api-time-attendance (tên là duy nhất)
  - OS: Windows
  - Publish: Code
  - Location: Southeast Asian
  - Pricing Tier: Standard S1 (CPU: 1 Core, Ram: 1.75 GB), chi phí 74.4\$/tháng
  - URL sau khi tạo: <https://api-time-attendance.azurewebsites.net>
- **Azure CDN:** Lưu trữ các file tĩnh như: Javascript, Css, html, ảnh.
  - Database name: time-attendance-cdn (tên là duy nhất)
  - Location: Southeast Asian
  - Pricing Tier: Standard Akamai
  - Tạo endpoint tới storage: <https://timeattendance.blob.core.windows.net/>
  - URL sau khi tạo thành công: <https://time-attendance-cdn.azureedge.net>

## b) Bố trí lắp đặt Camera

- Cấu hình Camera yêu cầu tối thiểu:
  - Độ phân giải 1Mpx (HD 1280x720)
  - Frame rate: 16 Frames per second
  - Chuẩn nén: H.264, H.265, MJPEG
  - Giao thức: RTSP, HTTP
- Vị trí lắp đặt: từ 1.6 đến 2.2m tính từ mặt đất tới vị trí lắp Camera
- Khoảng cách nhận diện chính xác nhất: từ 1.0 đến 4.5 m tính từ vị trí đứng tới Camera với điều kiện đủ ánh sáng, tránh ánh sáng chói chiếu vào mặt.

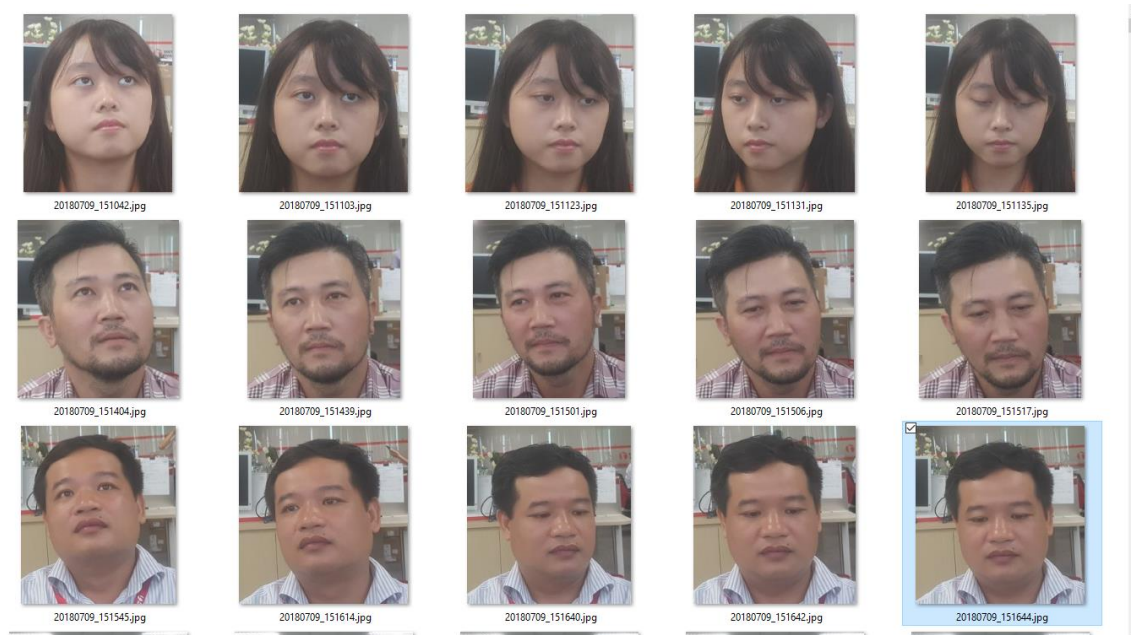


Hình 3.14: Hình ảnh bố trí Camera

**c) Khởi tạo dữ liệu ảnh nhân viên**

- Máy ảnh dùng để chụp ảnh có độ phân giải từ 2Mpx đến 8Mpx
- Khoảng cách chụp ảnh từ 0.8 đến 1.2m (tính từ vị trí chụp ảnh tới máy ảnh)
- Kích thước ảnh không quá 4MB
- Hình ảnh sau khi chụp yêu cầu rõ nét, không bị nhòe, mờ. Trong mỗi ảnh chỉ được phép có ảnh của 1 khuôn mặt
- Chụp ảnh với các góc chụp khác nhau (đảm bảo nâng cao tính chính xác khi nhân viên di chuyển). Các góc chụp nghiêng trái, nghiêng phải, ngang mặt, cúi mặt không quá 30 độ.



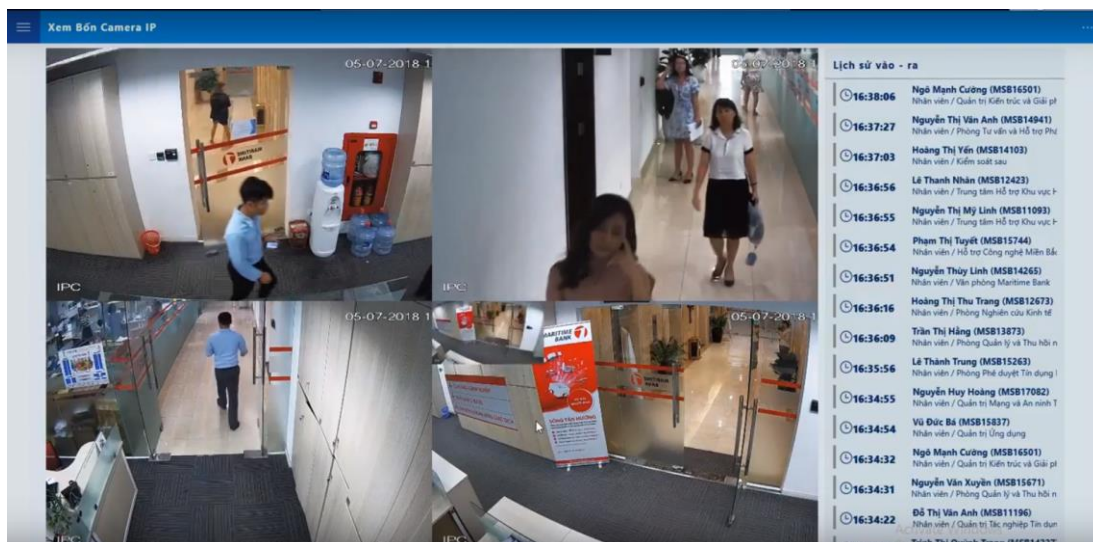


Hình 3.15: Ảnh mẫu nhận diện nhân viên

### 3.4.3. Màn hình giao diện

#### a) Giao diện theo dõi người vào ra

Mỗi khi có người đi qua vị trí đặt Camera, hệ thống sẽ thực hiện nhận diện dựa trên khuôn mặt. Sau khi nhận diện thành công, ngoài việc lưu vào thông tin vào ra, chấm công thì cũng hiển thị thông tin chấm công thành công lên màn hình cho nhân viên biết đã được chấm công



Hình 3.16: Giao diện theo dõi người vào ra

### b) Giao diện đăng nhập dành cho quản trị viên

Để có thể vào các giao diện quản lý, thì người dùng phải thực hiện đăng nhập vào hệ thống, việc này đảm bảo chỉ người được cấp quyền mới có thể truy cập và sử dụng các tính năng trong phần mềm.



Hình 3.17: Giao diện đăng nhập phần mềm

### c) Giao diện thông tin chấm công

Giao diện hiển thị thông tin chấm công của từng nhân viên theo thời gian. Người quản lý chấm công có thể tra cứu dựa trên thông tin về phòng ban, mã số nhân viên, thời gian chấm công,...

**PHẦN MỀM QUẢN LÝ CHẤM CÔNG**  
THÔNG TIN CHẤM CÔNG

**TIÊU CHÍ THỐNG KẾ**

CHỨC VỤ: << Tất cả >> PHÒNG BAN: << Tất cả >> MÃ NHÂN VIÊN: THỜI GIAN: 01/10/2018 - 07/10/2018 HO TÊN: CHON NĂM: Năm 2018

THỐNG KẾ: ☐ NGÀY ☒ TUẦN ☐ THÁNG

**THÔNG KẾ THỜI GIAN CHẤM CÔNG**

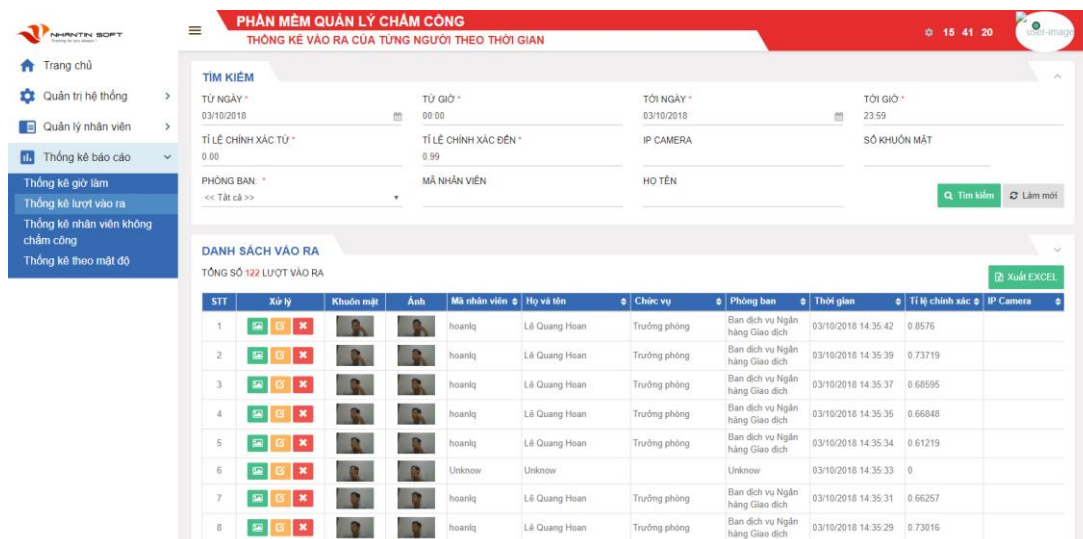
TỔNG SỐ 789 BẢNG GHI

STT	Ngày	Thứ	Khuôn mặt vào	Ảnh vào	Khuôn mặt ra	Ảnh ra	Mã nhân viên	Họ tên	Chức vụ	Phòng ban	Giờ vào
1	07/10/2018	Chủ Nhật					MSB12045	Nguyễn Đức Hưng	Nhân viên	Ngân hàng Doanh nghiệp Lớn	11:51
2	07/10/2018	Chủ Nhật					MSB06252	Phùng Thị Minh Diệp	Nhân viên	Trung tâm Hỗ trợ Khu vực Hà Nội	15:33
3	07/10/2018	Chủ Nhật					MSB15613	Hoàng Thị Minh Trang	Nhân viên	Trung tâm Kiểm soát Tội phạm (Hà Nội)	18:22
4	07/10/2018	Chủ Nhật					MSB11617	Phạm Quang Thanh	Nhân viên	Trung tâm Quản trị Tài chính	10:18
5	07/10/2018	Chủ Nhật					MSB14481	Vũ Đức Phương	Nhân viên	Ngân hàng Doanh nghiệp Lớn	15:36
6	07/10/2018	Chủ Nhật					MSB14132	Trần Ngọc Anh	Nhân viên	Phòng Quản lý Tài sản và mạng lưới	11:44
7	07/10/2018	Chủ Nhật					MSB14702	Đặng Thu Trang	Nhân viên	Trung tâm Hỗ trợ Khu vực Hà Nội	15:41
8	07/10/2018	Chủ Nhật					MSB14378	Bùi Thị Sơn	Nhân viên	Phòng Quản lý và Thu hồi nợ (Hà Nội)	15:44
9	07/10/2018	Chủ Nhật					MSR16478	Phạm Ngọc Đình	Nhân viên	Ran Duẩn từ Tin được	16:45

Hình 3.18: Giao diện thông tin chấm công

#### d) Giao diện thống kê lượt vào ra

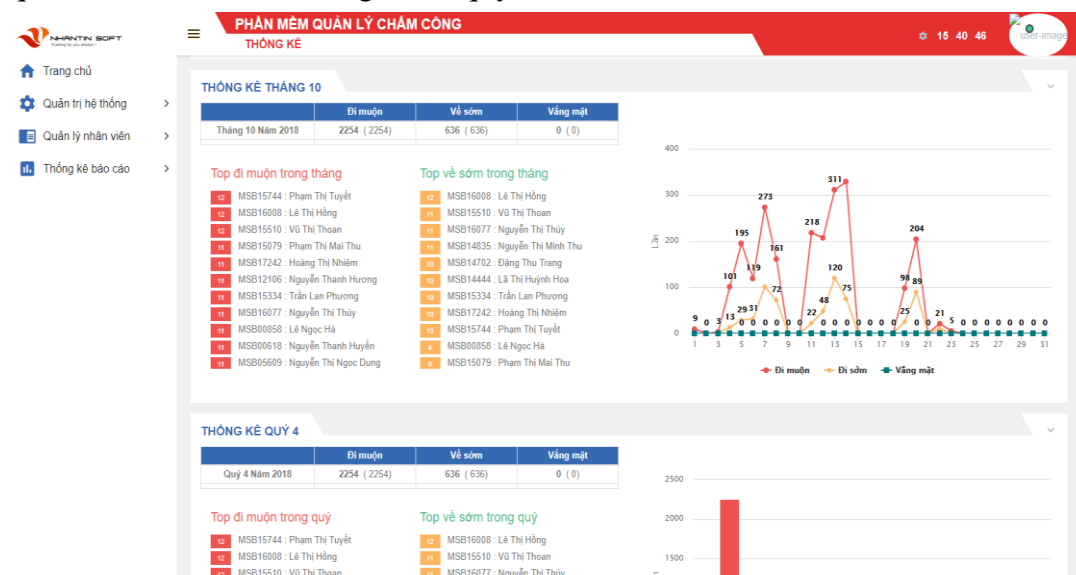
Tất cả các lượt vào, ra của nhân viên đều được hệ thống ghi nhận lại. Có thể tra cứu theo nhiều thông tin khác nhau như: khoảng thời gian xuất hiện, mã số nhân viên, tên nhân viên, vị trí lắp camera....



Hình 3.19: Giao diện thống kê lượt vào ra

#### e) Giao diện tổng hợp, thống kê

Giao diện tổng hợp, thống kê số lần đi làm muộn, về sớm của nhân viên trong cơ quan theo tuần, theo tháng, theo quý.

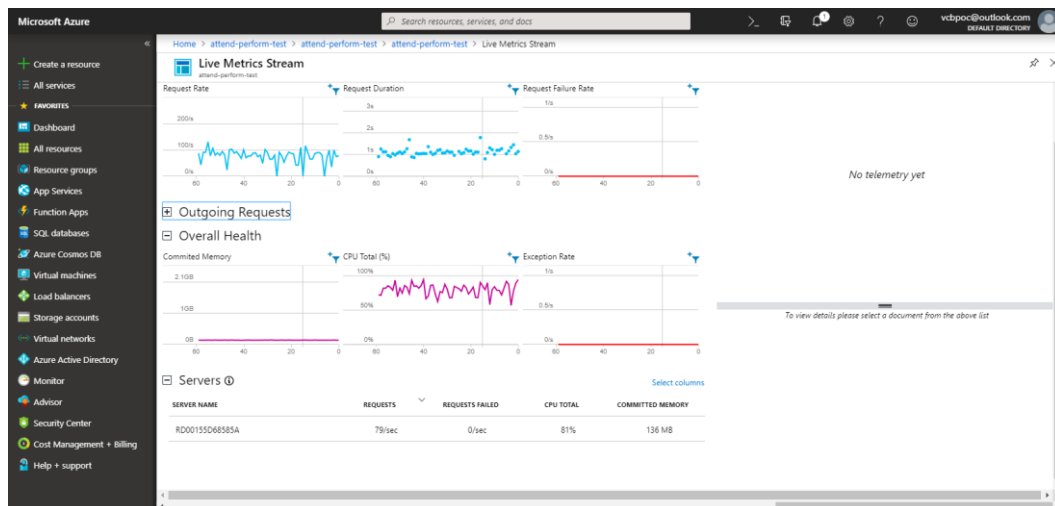


Hình 3.20: Giao diện tổng hợp, thống kê

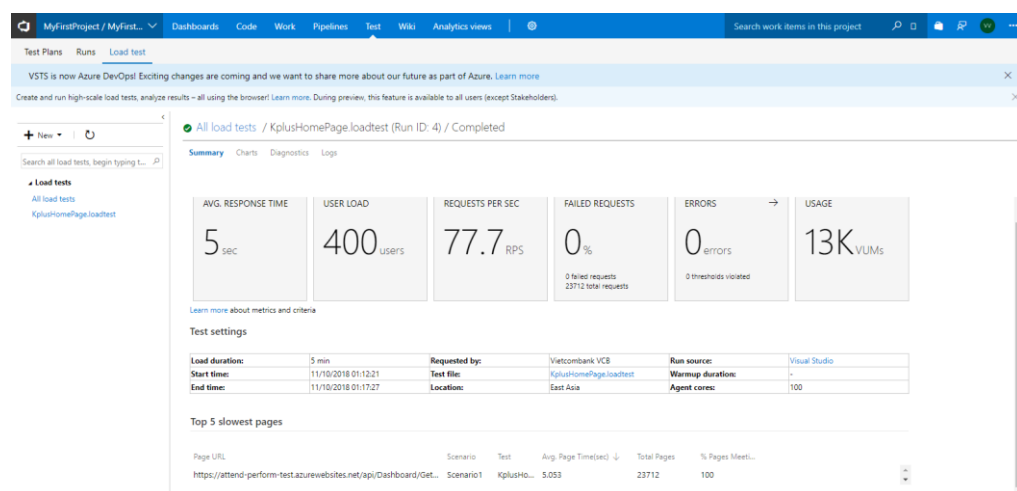
### 3.5. Đánh giá khả năng chịu tải của hệ thống

Đánh giá hiệu năng là một nhiệm vụ vô cùng quan trọng đối với các ứng dụng. Khi số lượng người dùng truy cập đồng thời một lúc tăng lên dẫn đến để đáp ứng được số lượng yêu cầu lớn hơn thì cần phải tăng số lượng các instance. Số lượng instances tăng đồng nghĩa với việc mỗi instance sẽ phải chịu một lượng tải nhỏ hơn, do đó hệ thống có thể đáp ứng nhiều người dùng hơn.

Để đánh giá hiệu năng của hệ thống, Microsoft cung cấp 2 công cụ đó là Azure Load Test và Azure Application Insight. Azure Load test cung cấp công cụ để thực hiện load test và đánh giá hiệu năng của hệ thống. Còn Application Insight cho phép xem các thông số về trạng thái CPU, mức độ sử dụng tài nguyên, Số lượng request đến theo thời gian thực.



Hình 3.21: Giao diện sử dụng Application Insight



Hình 3.22: Giao diện kết quả khi chạy load test

### 3.5.1. Đánh giá với số lượng user đồng thời tăng dần

- Cấu hình Web app:
  - o Ram: 1.75 Gb
  - o CPU: 1 Core
  - o Số lượng Instances: 1 instance
  - o API được test:

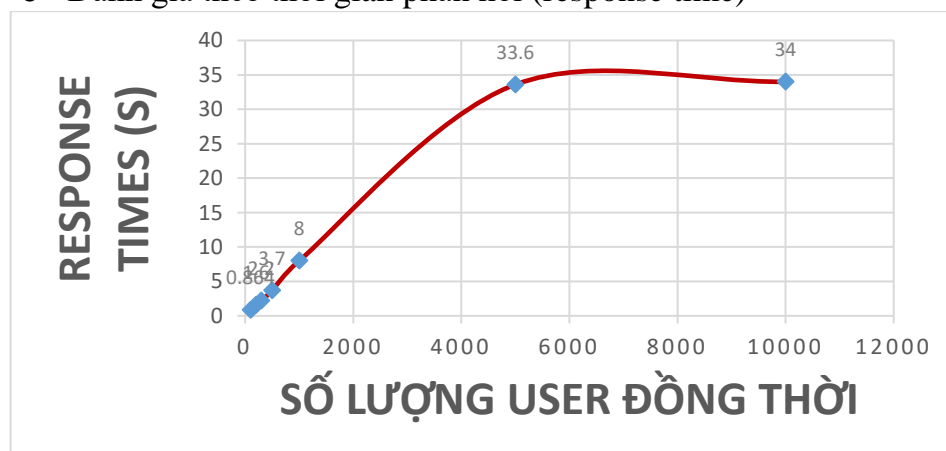
<https://api-time-attendance.azurewebsites.net/api/Dashboard/GetDataDashboard>

- User ở đây là user ảo được tạo ra từ hệ thống Azure Load test, các máy này sẽ tự động gửi yêu cầu tới trang thống kê của hệ thống.
- Kết quả sau khi thực hiện chạy test

STT	Số user	Response time (giây)	Số request / giây	% CPU	Ram used (Mb)
1	100	0.864	104.3	70	92 Mb
2	200	1.6	111.5	65	97 Mb
3	300	2.2	119.6	80	102 Mb
4	500	3.7	119.9	70	112 Mb
5	1000	8	109	66	144 Mb
6	5000	33.6	99.2	58	180 Mb
7	10000	34	106	60	240 Mb

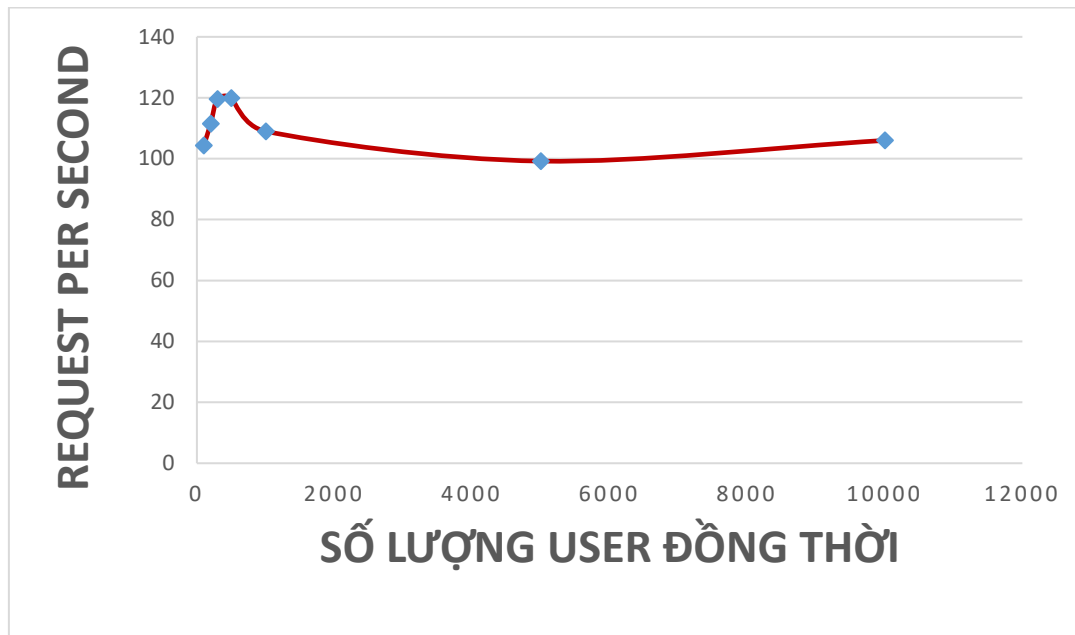
Bảng 3.1: Kết quả thực hiện load test 1

- Đánh giá kết quả:
  - o Đánh giá theo thời gian phản hồi (response time)



Hình 3.23: Biểu đồ số thể hiện thời gian thực hiện yêu cầu – test 1

- Đánh giá theo số request thực hiện được trong 1 giây



Hình 3.24: Biểu đồ số request thực hiện được trong 1 giây – test 1

Kết quả thực hiện cho thấy khi số lượng người dùng đồng thời cùng một lúc tăng, cấu hình của server không đổi nên lượng yêu cầu có thể xử lý đồng thời cùng một lúc không thay đổi nhiều. Khi số lượng yêu cầu đến càng nhiều, làm cho thời gian phản hồi lại các yêu cầu này ngày càng tăng.

### 3.5.2. Kiểm thử với số lượng instance tăng dần

- Cấu hình Web app:
  - Ram: 1.75 Gb
  - CPU: 1 Core
  - Số lượng user đồng thời: 10.000 user
- User ở đây là user ảo được tạo ra từ hệ thống Azure Load test, các máy này sẽ tự động gửi yêu cầu tới trang thống kê của hệ thống.
- API được test:

<https://api-time-attendance.azurewebsites.net/api/Dashboard/GetDataDashboard>

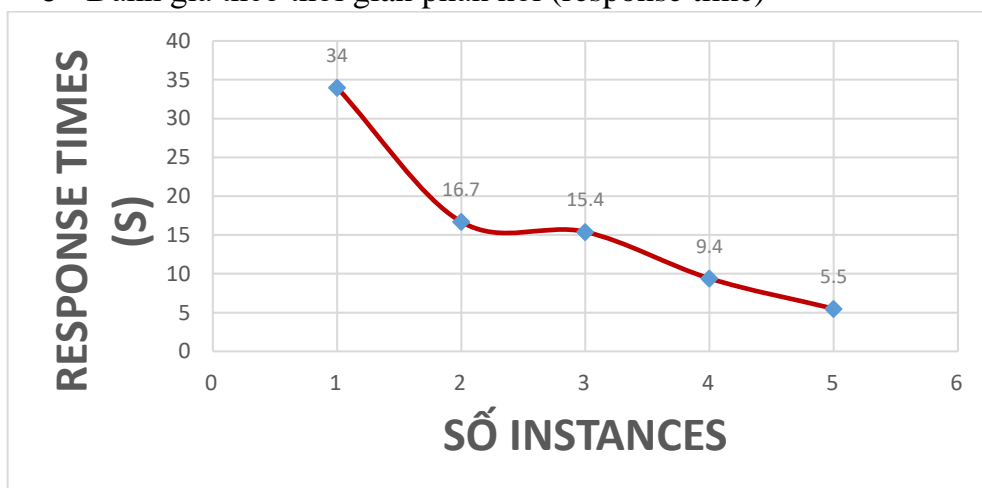


- Kết quả sau khi thực hiện chạy test

STT	Số user	Số instances	Response time (giây)	Số request / giây	% CPU	Ram used
1	10000	1	34	106	60	240 Mb
2	10000	2	16.7	222.6	65	220 Mb
3	10000	3	15.4	263.3	70	180 Mb
4	10000	4	9.4	440	72	150 Mb
5	10000	5	5.5	600	66	120 Mb

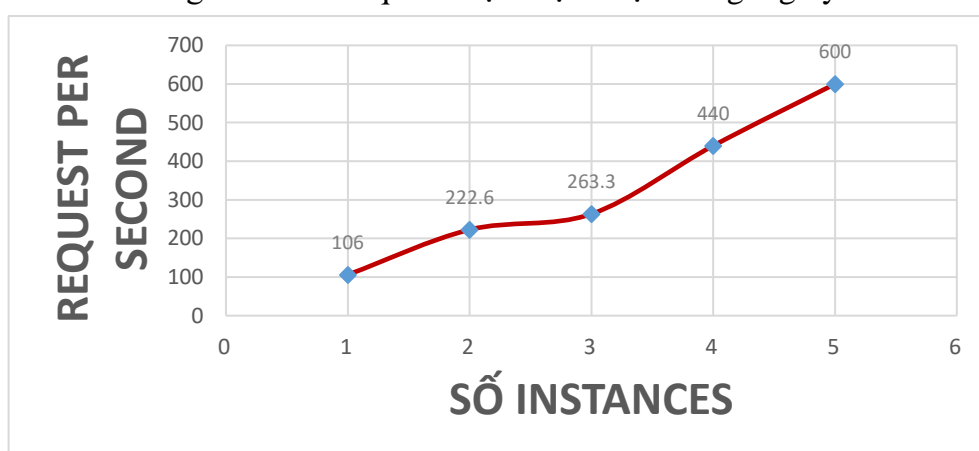
Bảng 3.2: Kết quả thực hiện load test 2

- Đánh giá kết quả:
  - o Đánh giá theo thời gian phản hồi (response time)



Hình 3.25: Biểu đồ số thể hiện thời gian thực hiện yêu cầu – test 2

- o Đánh giá theo số request thực hiện được trong 1 giây



Hình 3.26: Biểu đồ số request thực hiện được trong 1 giây – test 2

Kết quả thực hiện cho thấy, với cùng một số lượng người dùng, khi ta tăng số lượng instances, số lượng yêu cầu sẽ được san cho các instance cùng xử lý. Nhờ việc tăng số lượng instance mà thời gian phản hồi lại với mỗi yêu cầu giảm, do đó số lượng yêu cầu được xử lý trong 1 giây tăng. Do đó để tăng khả năng chịu tải của hệ thống, thì cách tốt nhất là tăng số lượng instance lên.

### **3.6. Kết luận**

Như vậy chương này đã giới thiệu về bài toán chấm công bằng nhận diện khuôn mặt và quá trình triển khai ứng dụng lên môi trường Microsoft Azure, thử nghiệm quy trình hệ thống với các tính năng như khởi tạo dữ liệu khuôn mặt ban đầu, nhận diện nhân viên khi ra vào tại các vị trí đặt camera quan sát, thống kê thời gian chấm công của từng nhân viên. Hệ thống hiện đang được triển khai ở trụ sở chính của ngân hàng Thương mại cổ phần Hàng Hải (Maritime bank) tại tầng 28, tòa nhà TNR, 54 Nguyễn Chí Thanh, Phường Láng Thượng, Quận Đống Đa, Hà Nội với số lượng nhân viên là 481 người.

Chương này cũng đã thực hiện thử nghiệm khả năng chịu tải của hệ thống với số lượng lớn người dùng đồng thời cùng một thời điểm. Với cấu hình của hệ thống ở mức thấp (CPU 1 core, Ram 1.75 Gb), chạy với 5 instance và có 10.000 người đồng thời thì hệ thống vẫn có khả năng xử lý được 600 request/ 1 giây. Do đó khi lựa chọn cấu hình hệ thống với các mức cao hơn, số lượng instance nhiều hơn thì hệ thống hoàn toàn có thể đáp ứng được nhiều hơn nữa số lượng người dùng đồng thời cùng một lúc.



## Chương 4. KẾT LUẬN

Việc triển khai các ứng dụng trên nền tảng điện toán đám mây dần trở thành một xu hướng tất yếu nhờ các ưu điểm vượt trội của điện toán đám mây. Việc triển khai ứng dụng trên nền tảng đám mây giúp cho doanh nghiệp tiết kiệm được khoản đầu tư ban đầu tương đối lớn về cơ sở hạ tầng. Với khả năng co giãn về kích cỡ và việc tính chi phí theo thực dùng, doanh nghiệp không phải lo lắng về việc lãng phí tài nguyên khi có biến động về nhân sự.

Sau thời gian tìm hiểu, nghiên cứu tài liệu và làm luận văn dưới sự hướng dẫn của thầy TS. Trần Trọng Hiếu và thầy PGS-TS. Phạm Ngọc Hùng tôi đã hoàn thành luận văn với đề tài “**Kiến trúc phần mềm chịu tải cao dựa trên nền tảng điện toán đám mây Microsoft Azure**”. Luận văn đã đạt được kết quả sau:

- Tìm hiểu, nghiên cứu những lý thuyết tổng quan liên quan đến các dịch vụ trên điện toán đám mây.
- Tìm hiểu về các dạng kiến trúc phần mềm triển khai trên nền tảng đám mây Microsoft Azure
- Xây dựng thành công ứng dụng trên nền tảng đám mây và đánh giá khả năng chịu tải của ứng dụng sau khi được triển khai.

Kết quả nghiên cứu của luận văn có thể áp dụng cho các dự án về chuyển đổi hệ thống cho các ứng dụng chạy ở on-premise lên nền tảng cloud, hoặc các dự án phát triển mới trên nền tảng Microsoft Azure có khả năng đáp ứng lượng lớn người dùng truy cập đồng thời.

Hướng phát triển trong tương lai của đề tài:

- Xây dựng thêm các chức năng trên ứng dụng Mobile cho phép nhân viên đăng nhập để xem thông tin chấm công, đăng ký lịch nghỉ,...
- Tích hợp với hệ thống tính tiền lương để tính lương cho nhân viên
- Phát triển phần nhận diện hình ảnh trên các thiết bị nhỏ gọn (Raspberry)

Trong quá trình nghiên cứu và thực hiện, học viên đã nhận được sự giúp đỡ nhiệt tình của các thầy hướng dẫn TS. Trần Trọng Hiếu, PGS-TS. Phạm Ngọc Hùng và các thầy, cô trong Khoa Công nghệ thông tin, Trường Đại học Công nghệ, ĐHQG Hà Nội. Học viên xin chân thành cảm ơn các thầy cô, và xin kính chúc các thầy cô luôn luôn mạnh khỏe và hạnh phúc !

## TÀI LIỆU THAM KHẢO

- [1] Ashraf S. Huwedi, “Preprocessing Techniques for Face Recognition using R-LDA under Difficult Lighting Conditions and Occlusions”
- [2] Bakhshi, Yukti, Sukhvirkaur, and Prince Verma. "An Improvement in Face Recognition for Invariant Faces.", 2016
- [3] Bill Wilder, “Cloud Architecture Pattern”, O'Reilly Media, 2012
- [4] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen and Zhenghu Gong, “The Characteristics of Cloud Computing”, 2010
- [5] David Chappel, “Introducing Windows Azure”, 2009
- [6] E.R. Davies, “Computer and Machine Vision: Theory, Algorithms Practicalities”, fourth edition, 2012
- [7] Huda M.S. Algharib, "Face Recognition under Difficult Lighting Conditions and Occlusions", ICIPCS, 2015
- [8] K. Dharavath, G. Amarnath, F.A. Talukdar, and R.H. Laskar. “Impact of image preprocessing on face recognition: A comparative analysis”, ICCSP, 2014
- [9] Len Bass, Paul Clements, Rick Kazman: “Software Architecture in Practice”, third edition, Addison-Wesley, 2013
- [10] Michael Collier, Robin Shahan, “Microsoft Azure Essentials – Fundamentals of Azure”, Microsoft Press, 2015
- [11] Mike Wasson, Masashi Narumoto, “Cloud Application Architecture Guide”, Microsoft Press, 2017
- [12] Pavel Rabetski, Gerardo Schneider, “Migration of an on-premise application to the Cloud”, Service-Oriented and Cloud Computing, Springer Berlin Heidelberg, pp. 227-241, 2013
- [13] Parmar, Divyarajsinh N., and Brijesh B. Mehta. "Face Recognition Methods & Applications.", 2014

- [14] Peter Mell, Timothy Grance, “The NIST Definition of Cloud Computing”, Communications of the ACM. 53, 2011
- [15] Rashmi Chaurasiya, Surabhi Varshney, Yogesh Tayal, “Image Processing Techniques for Face Recognition Application”, International Journal of Engineering and Technical Research, 2014
- [16] Roy Kim, “Microsoft Azure Cloud for Solution Architects”, 2015
- [17] Steve Smith, “Architecting Modern Web Applications with ASP.Net Core and Microsoft Azure”, Microsoft Press, 2017
- [18] <https://azure.microsoft.com/>