INDIVIDUAL ASSIGNMENT

CT071-3-3-DDAC

DESIGNING & DEVELOPING CLOUD APPLICATIONS

UC3F1610IT(ISS)

HAND IN DATE: 14ST JULY 2017

STUDENT'S NAME: LAU KOK LI

STUDENT ID: TP031874

LECTURER'S NAME: DR. KALAI ANAND RATNAM

Acknowledgment

There are several people I would like to show my appreciation as they have provided a lot useful recommendation and guidance throughout the entire project. This Container Management System project would not have successfully accomplished without supportive advice while I was facing challenges during the development and deployment of system.

First, I would like to thank Dr. Kalai Anand A/L Ratnam, my lecturer who provided knowledge and guidance regarding to cloud deployment. He had provided tutorial as well as clear explanation in order to have better understanding about the use of Microsoft Azure technology.

Next, I would like to thank my friends who had gave valuable recommendation for simplifying the system development. We have shared all the knowledge gained in order to enhance system development and deployment process.

Github projet: https://github.com/NoobProgrammer/cms031874

System portal: http://cms031874.azurewebsite.net/ (Southeast Asia)

http://cms031874ne.azurewebsite.net/ (North Europe)

Account: entadmin@koklilau95hotmail.onmicrosoft.com

Password: wfFF2zdP

Table of Contents

Introduction

Background

Maersk Line is the global container division and the largest operating unit of the A.P. Moller – Maersk Group, a Danish business conglomerate. It is the world's largest container shipping company having customers through 374 offices in 116 countries. It employs approximately 7,000 sea farers and approximately 25,000 land-based people. Maersk Line operates over 600 vessels and has a capacity of 2.6 million TEU. The company was founded in 1928.

Operating in 100 countries and transporting goods around the globe, at first glance it would appear Danish shipping company Maersk Line is already handling all the cargo it can manage. But when Maersk determined that the volume of most of the goods it was shipping had grown to full capacity, the company decided that cloud powered solutions would be a crucial part of rectifying the situation.

“There was a ‘mind-opener’ where Maersk said, ‘How can we support the overall business strategy, and also from an IT perspective,” says Soeren Lorenzen, an account general manager with Hewlett-Packard company who is involved first-hand with Maersk’s ITO efforts. “There was a new CIO who wanted to outsource every part of IT, but without [negatively] impacting shipping.”

In an effort to support further business growth and increase organizational flexibility, Maersk decided to consolidate all of its data centres and server rooms operating worldwide onto a virtualized platform. Microsoft Azure was already hosting some of Maersk’s IT environment, and in March 2016 Maersk initially approached Microsoft about expanding the scope of the relationship. Moving forward, Lorenzen says Maersk is currently changing over its IT setup based on Microsoft Azure, starting with the desktop environment up to container management.

Objectives

To development a Container Management System with cloud supportive solution that reduces overall supply chain costs and an efficient way to manage shipments.

Scope

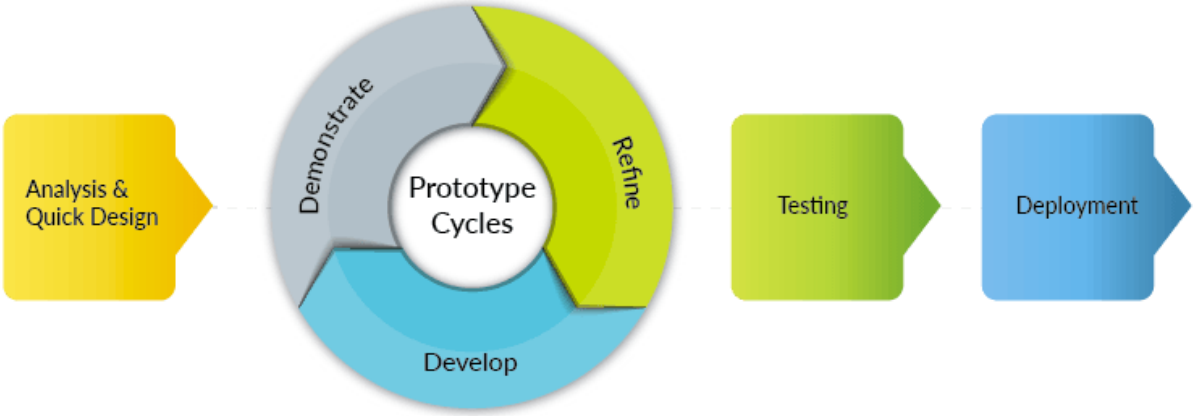
1. From import, export and transshipment processing to gate operations.
2. To be able to scale the solution to meet the needs of demands during peak seasons.
3. Improves profitability, cuts costs, increases productivity, eradicates errors and optimizes resources to future-proof your cargo handling business for high performance.
4. Assurance & reliability through Failover Management.
5. Accurately allocates inbound containers to yard locations and plan outbound containers to individual hauler vehicles, delivering an exceptional level of automation and removing human error.
6. Manage your entire booking process from schedule search to booking confirmation
7. Able to handle high load request in the same time

Requirement Specification

- ✓ ***Provisioning:*** You must be able to provision the new application to the Microsoft Azure Platform.
- ✓ ***Maintainability:*** You must be able to upgrade the application and perform other maintenance tasks while multiple tenants are using it.
- ✓ ***Monitoring:*** You must be able to monitor the application at all times to identify any problems and to troubleshoot them. This includes monitoring how each tenant is using the application.
- ✓ ***Availability:*** Tenants want the application to be constantly available, perhaps with guarantees defined in an SLA. Again, the activities of other tenants should not affect the availability of the application.
- ✓ ***Scalability:*** The application scales to meet the demand of the application.

Project Plan

Methodology



Rapid Application Development (RAD) methodology is used for this project of system development. It relies on prototype cycle which able to speed up its development by getting feedback from user. RAD methodology characterized by reiterative user testing and reuse of software components (WaveMaker, 2017) which has reduced the redundancy of development process as well as avoided unnecessary bugs occur in the future. There are 4 phases in RAD methodology which are Analysis & Quick Design, Prototype Cycle, Testing and Development.

Analysis & Quick Design

In the first phase, developer conducts a research in order to gather the requirement of container management system. Regarding to the problem identification, quantitative and qualitative data will be collected through questionnaires and interviews. Then, collected data will be analysed and provided the best solution and requirement for system. Early stage of the system design will be drawn with Unified Modelling Languages (UML) in order to have a clear picture of system development.

Prototype Cycle

In the second phase, system development has started based on the design drawn in previous stage. The prototype of the system is developed and demonstrated to user in order to get the feedback of system. Modification and refinement of system features are to be done to meet the user requirements and satisfaction.

Testing

In this phase, system goes through various type of testing such as unit testing, integration testing, performance testing and so on. These testing is to ensure overall functionalities of system developed are flawless while it has been released to the targeted users. The purpose of testing is to improve the quality of system and minimize the errors or bugs occur.

Deployment

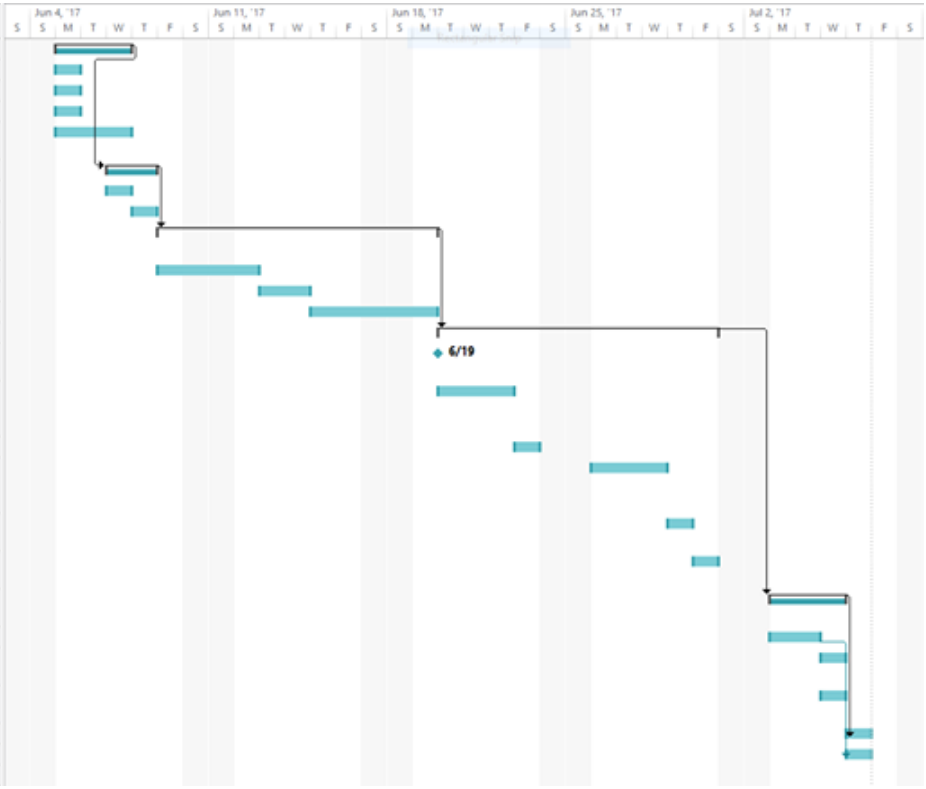
In the final phase, the system is finalized and prepared to be released. The cutover plan will be created by the organization for moving to new system from old system. Developer follows the cutover plan to deploy container management system.

Justification

The reason of using RAD methodology for this project is to reduce the great amount of time consumption whereby the formality of requirement documentation in planning phase consume a lot of time. In addition, the reusability of system components has increase the effectiveness of system development which is the web application. RAD methodology uses prototype in development process which can be reduce the cost consumption for the project. Developer develops the system based on prototype has reduced time and cost consumption which is significant to the core of project.

Gantt Chart

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Introduction**	**3 days**	**Mon 6/5/17**	**Wed 6/7/17**		
2		Background	1 day	Mon 6/5/17	Mon 6/5/17		
3		Objective	1 day	Mon 6/5/17	Mon 6/5/17		
4		Scope	1 day	Mon 6/5/17	Mon 6/5/17		
5		Requirement Specification	3 days	Mon 6/5/17	Wed 6/7/17		
6		**Project Plan**	**2 days**	**Wed 6/7/17**	**Thu 6/8/17**	**1**	
7		Methodology	1 day	Wed 6/7/17	Wed 6/7/17		
8		Gantt Chart	1 day	Thu 6/8/17	Thu 6/8/17		
9		**Design & Solution Architecture**	**7 days**	**Thu 6/8/17**	**Mon 6/19/17**	**6**	
10		Architecture design	2 days	Thu 6/8/17	Mon 6/12/17		
11		Design consideration	2 days	Tue 6/13/17	Wed 6/14/17		
12		Modelling	3 days	Thu 6/15/17	Mon 6/19/17		
13		**Implementation**	**9 days**	**Mon 6/19/17**	**Fri 6/30/17**	**9**	
14		Configure Primary & Secondary Database	0 days	Mon 6/19/17	Mon 6/19/17		
15		Configure secure access in Web Application with	3 days	Tue 6/20/17	Thu 6/22/17		
16		Publish Web Applicati	1 day	Fri 6/23/17	Fri 6/23/17		
17		Configure secure access with Azure Active Directory	3 days	Mon 6/26/17	Wed 6/28/17		
18		Configure Traffic Manager	1 day	Thu 6/29/17	Thu 6/29/17		
19		Enhance Performance with Application	1 day	Fri 6/30/17	Fri 6/30/17		
20		**Test Plan and Testing Discussion**	**3 days**	**Mon 7/3/17**	**Wed 7/5/17**	**13**	
21		Test Plan	2 days	Mon 7/3/17	Tue 7/4/17		
22		Web Application Performance Analysis	1 day	Wed 7/5/17	Wed 7/5/17		
23		Health Analysis with Application Insights	1 day	Wed 7/5/17	Wed 7/5/17		
24		Conclusion	1 day	Thu 7/6/17	Thu 7/6/17	20	
25		Reference	1 day	Thu 7/6/17	Thu 7/6/17	21	



Design & Solution Architecture

Architectural design

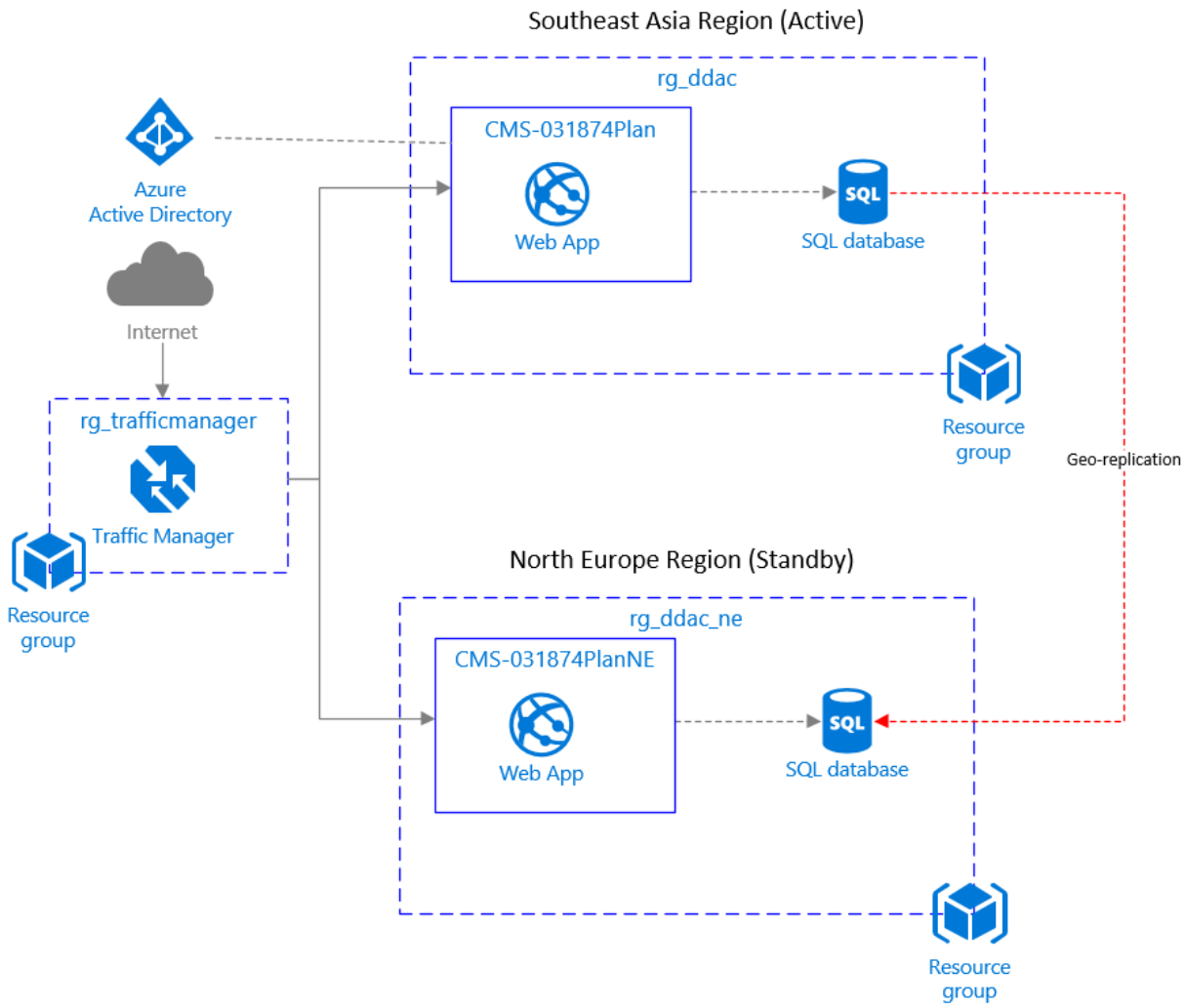


Figure 1: Azure Architecture

Design consideration

Simple database

Container Management System is designed for staff administrator to manage shipments. The simplicity of database would enhance the overall performance of web application synchronization to cloud database.

User-friendly web application interface

The user-friendly web application interface increase the productivity of staff administrator as he/she does not require any technical knowledge or user manual while using the system.

Security of web application / system

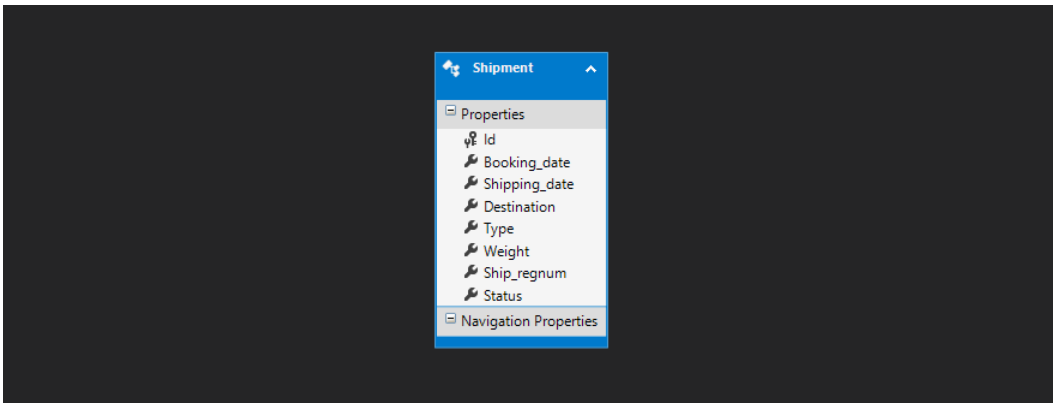
The login access is important to business which Azure Active Directory is included into the project for security purpose as well as increase the range of access management across different devices.

Scalability of web application

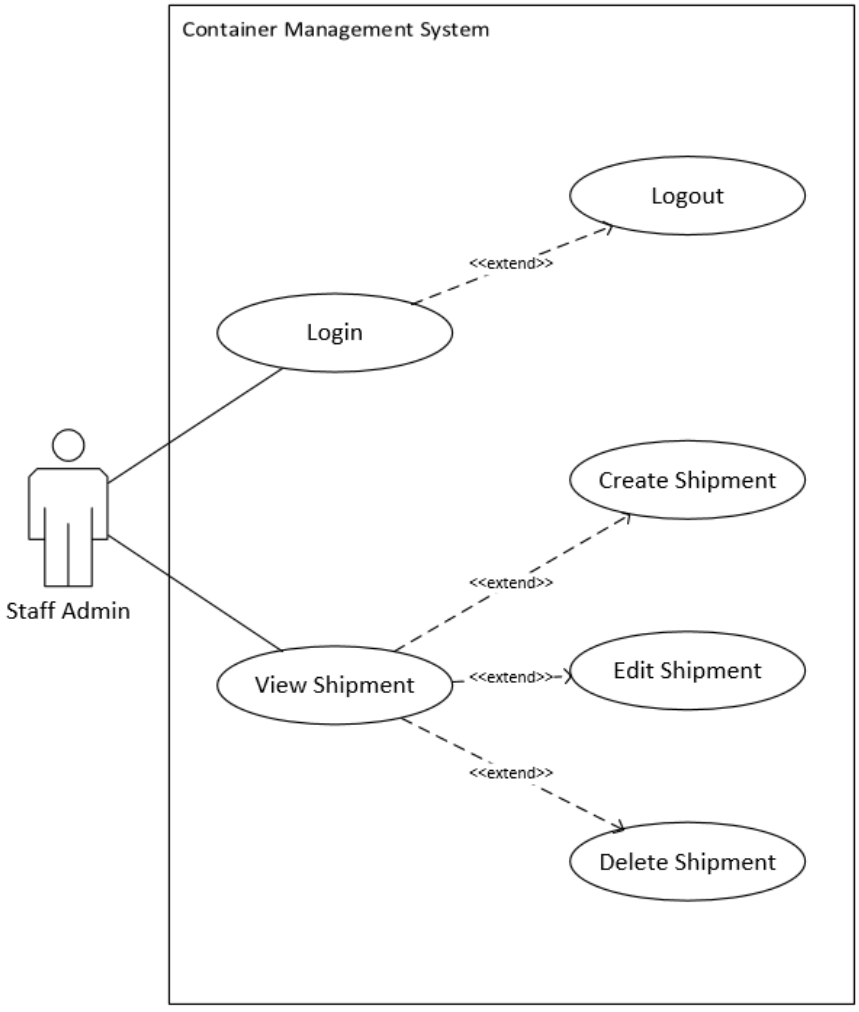
The App Service plan featured with scale up and scale out which significantly beneficial to current business project while can adjust the user load handling with autoscaling component.

Modelling

- Entity Relation Diagram (ERD)



- Use case diagram



- Use case specification

Name	Login	
Summary	Staff administrator login	
Priority	High	
Actor(s)	Staff administrator	
Pre-conditions	Staff administrator must have an account	
Main Flow	Step	Action
	1	Enter username and password
	2	Verifying username and password with active directory
	3	After verified, redirect to system page.

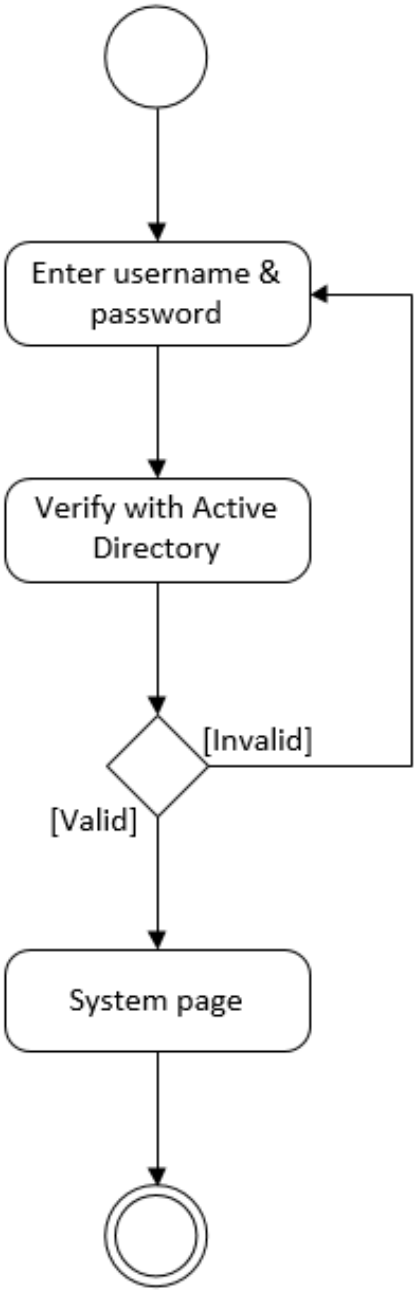
Name	View Shipment	
Summary	Staff administrator	
Priority	High	
Actor(s)	Staff administrator	
Pre-conditions	Staff administrator must have logged-in	
Main Flow	Step	Action
	1	Show entire list of shipments made

Name	Create Shipment	
Summary	Staff administrator chooses “Create Shipment”	
Priority	High	
Actor(s)	Staff administrator	
Pre-conditions	Staff administrator must have logged-in	
Main Flow	Step	Action
	1	Enter new shipment details
	2	Insert new shipment into database

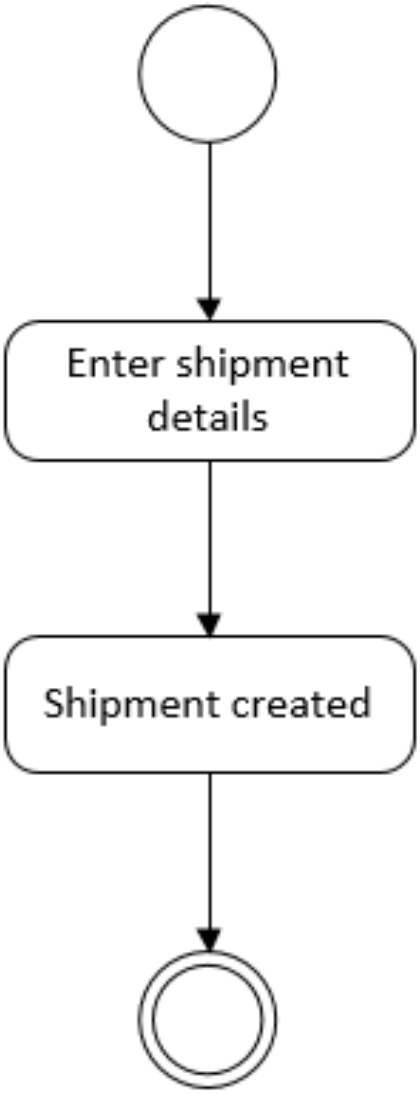
Name	Edit Shipment	
Summary	Staff administrator chooses “Edit”	
Priority	High	
Actor(s)	Staff administrator	
Pre-conditions	Staff administrator must have logged-in	
Main Flow	Step	Action
	1	Show selected shipment details
	2	Make any updates
	3	Save edited updates into database

Name	Delete Shipment	
Summary	Staff administrator chooses “Delete”	
Priority	High	
Actor(s)	Staff administrator	
Pre-conditions	Staff administrator must have logged-in	
Main Flow	Step	Action
	1	Delete selected shipment
	2	Refresh entire list of shipments

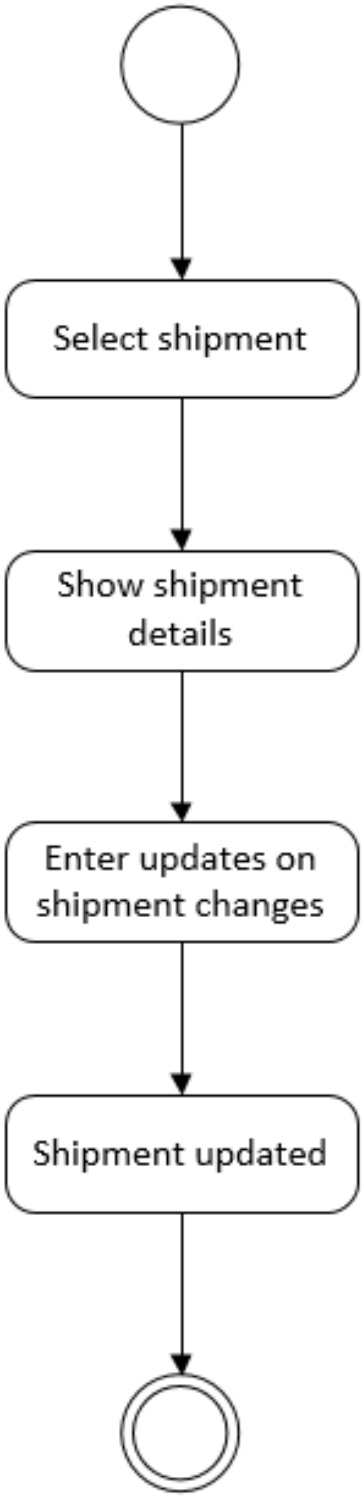
- Activity diagram
    - User login



- Create shipment



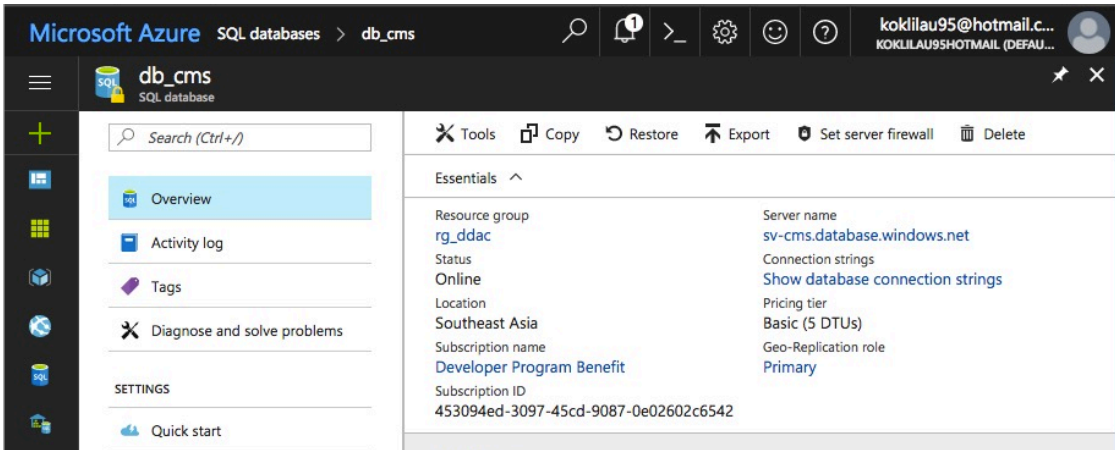
- Edit shipment



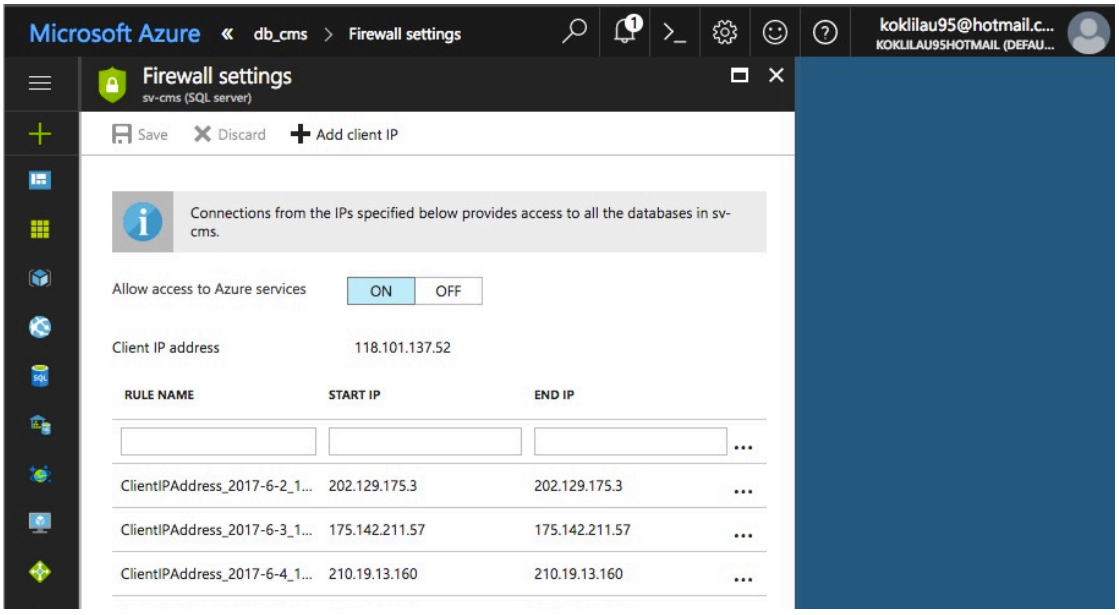
Implementation

Configure Primary and Secondary Database

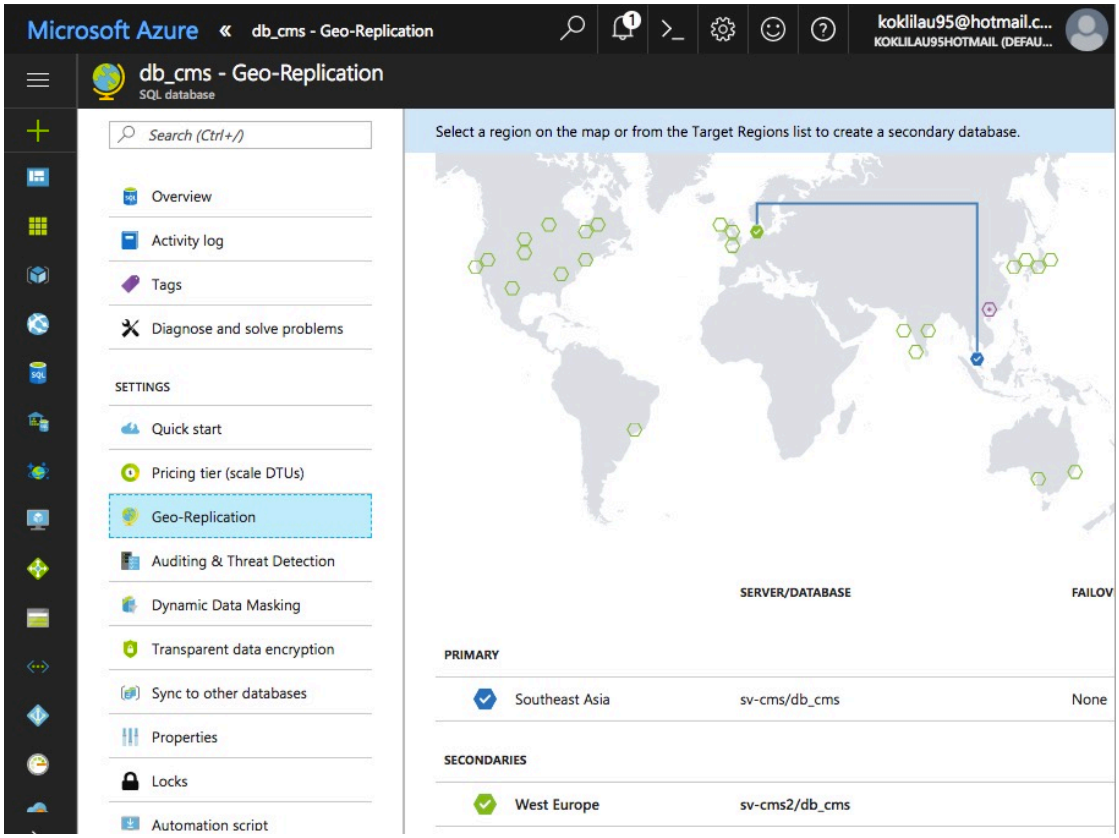
- In Azure portal, create a SQL database (db_cms) within resource group (rg_ddac) and choose the region of Southeast Asia.



- Then, choose “set server firewall” in order to accept connection from local machine which adding the personal IP address to access SQL server.

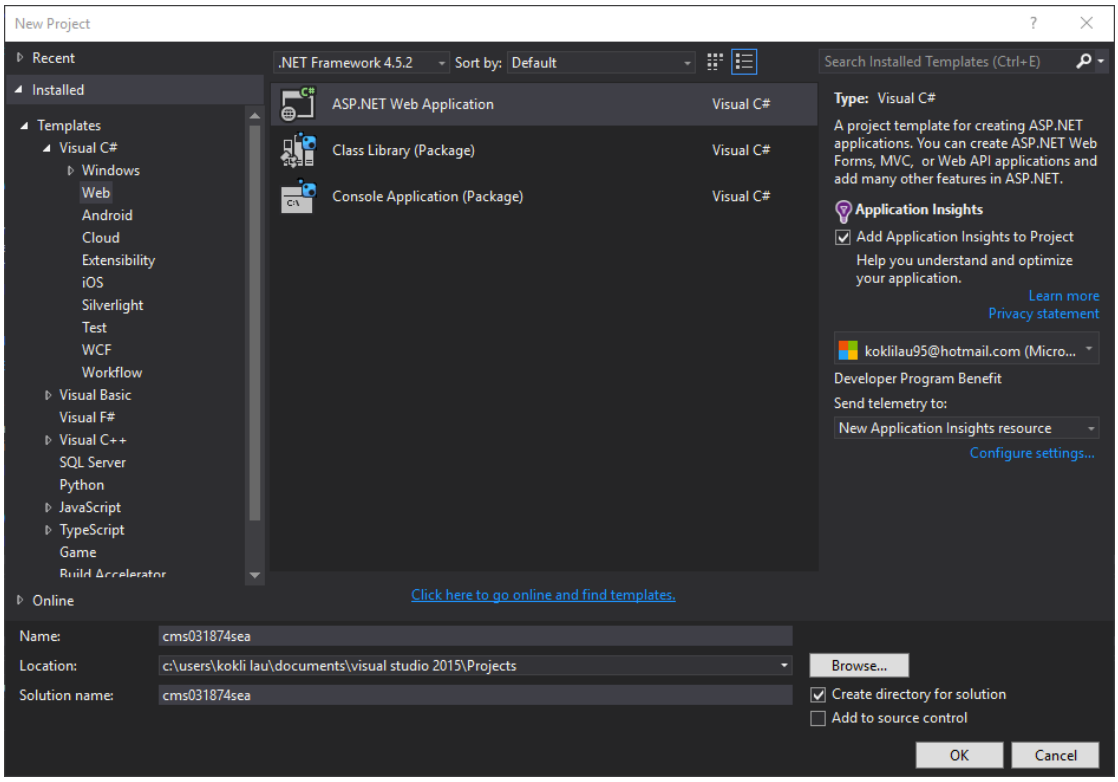


- Secondary database is created through “Geo-Replication” in order to overcome the failover web application. The connection between both databases will be shown on the map. The secondary database will remain read-only status until the primary database has failed over.

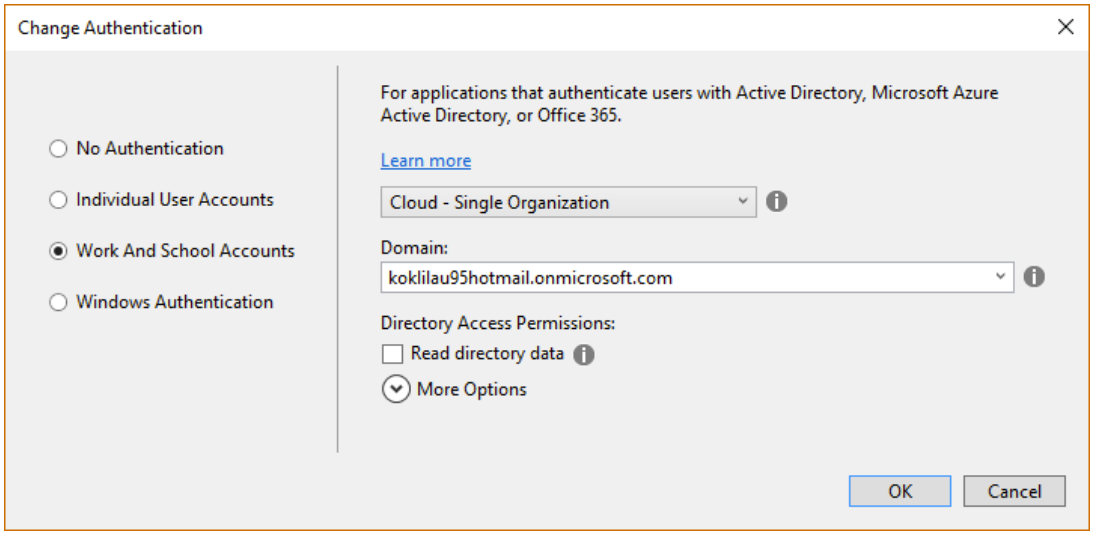


Configure secure access in Web Application with Active Directory

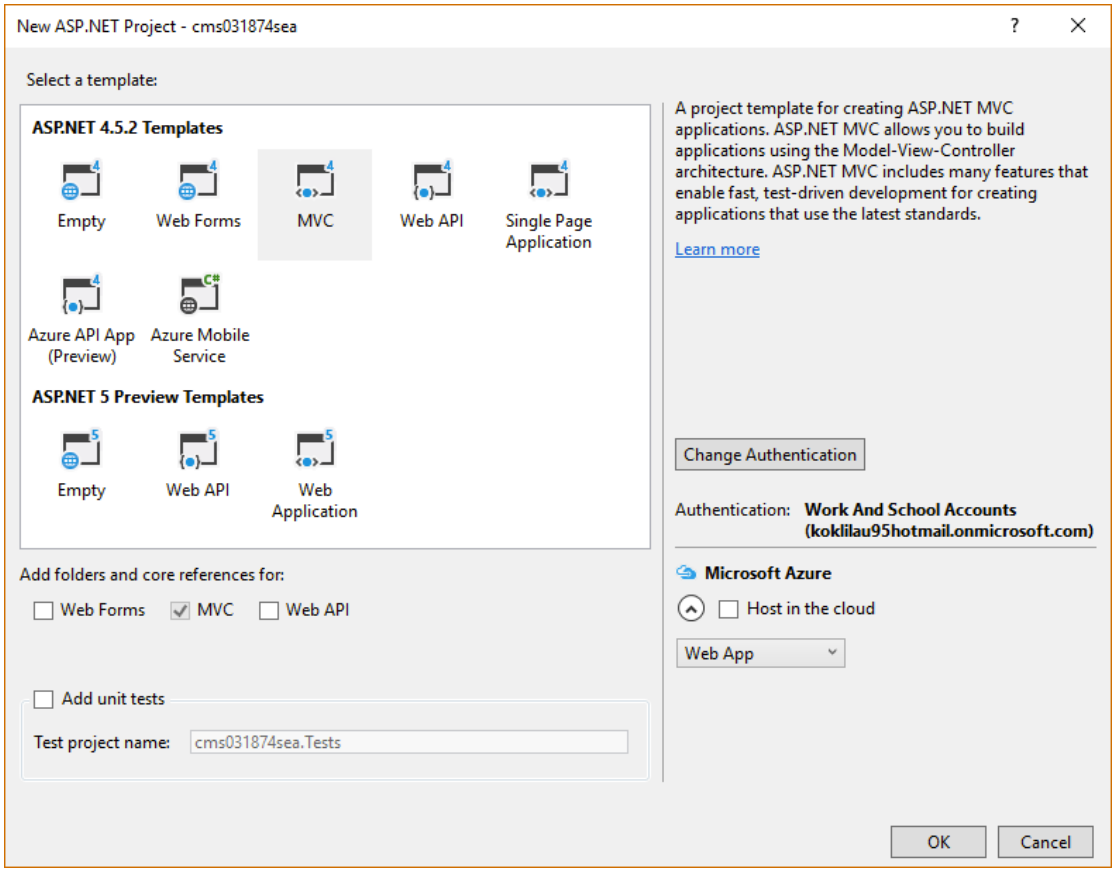
- In visual studio, create ASP.NET web application.



- Then, select “Change Authentication” and choose “Work And School Accounts” option along with entering account domain into the textbox given.

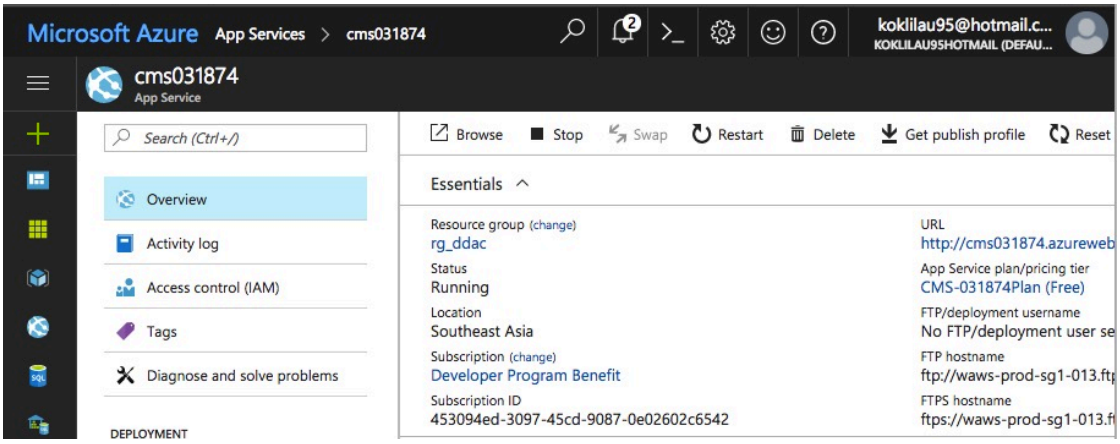


- Lastly, make sure the “MVC” template is selected which give convenience to developer while creating the components.

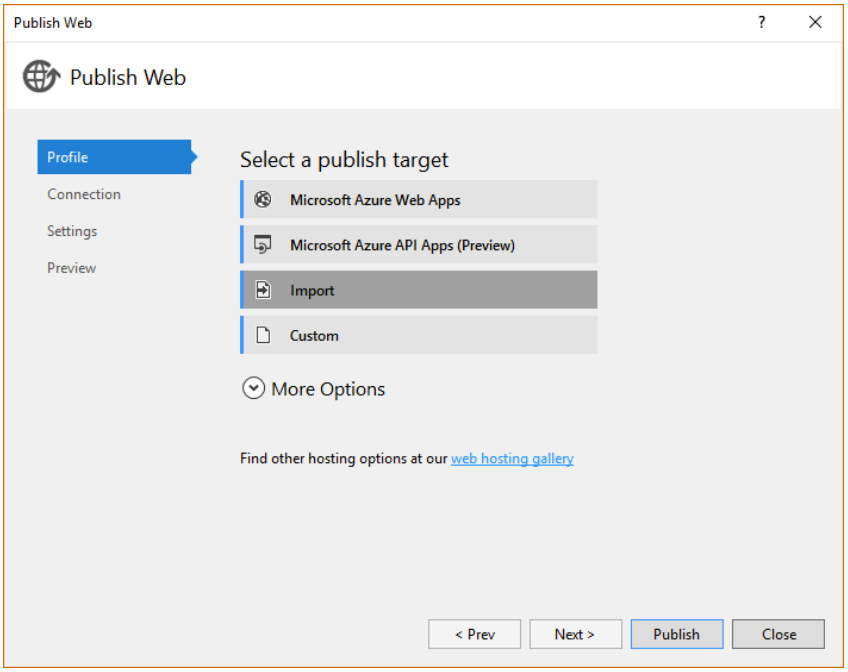


Publish Web Application

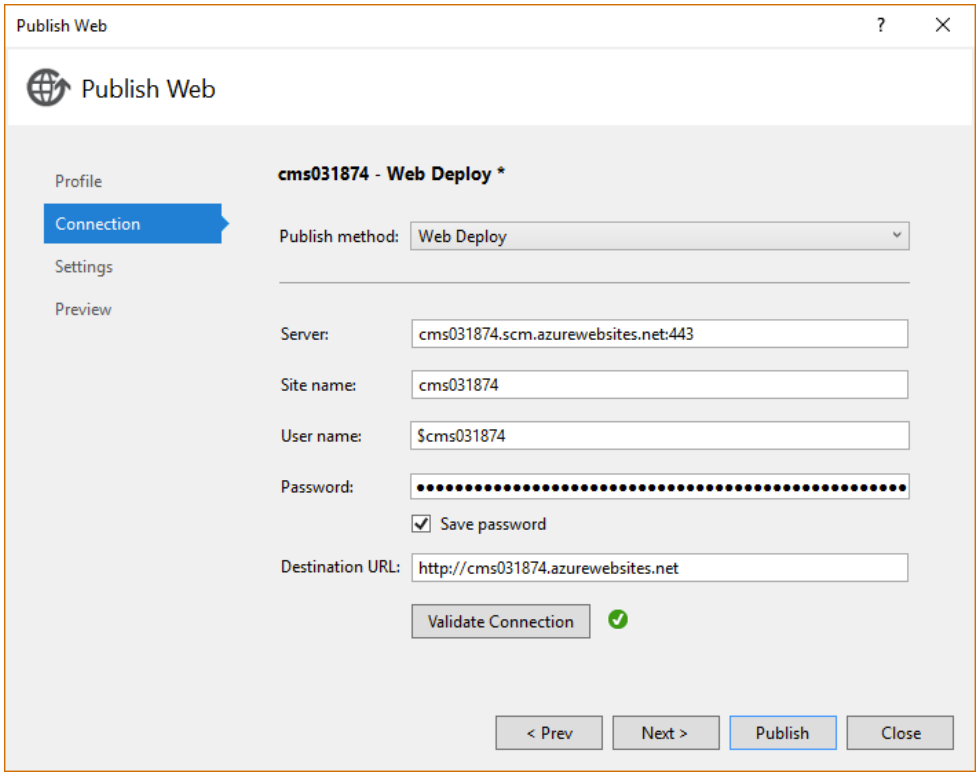
- In Azure portal, create two resource groups (rg_ddac) and (rg_ddacne) which indicate Southeast Asia region and North Europe region.
- Then, create web application through "App Service" in each of resource groups. The service plan must be using S1 for the further usage in next section.
- After web application is created, select "Get publish profile" for the manual publish.



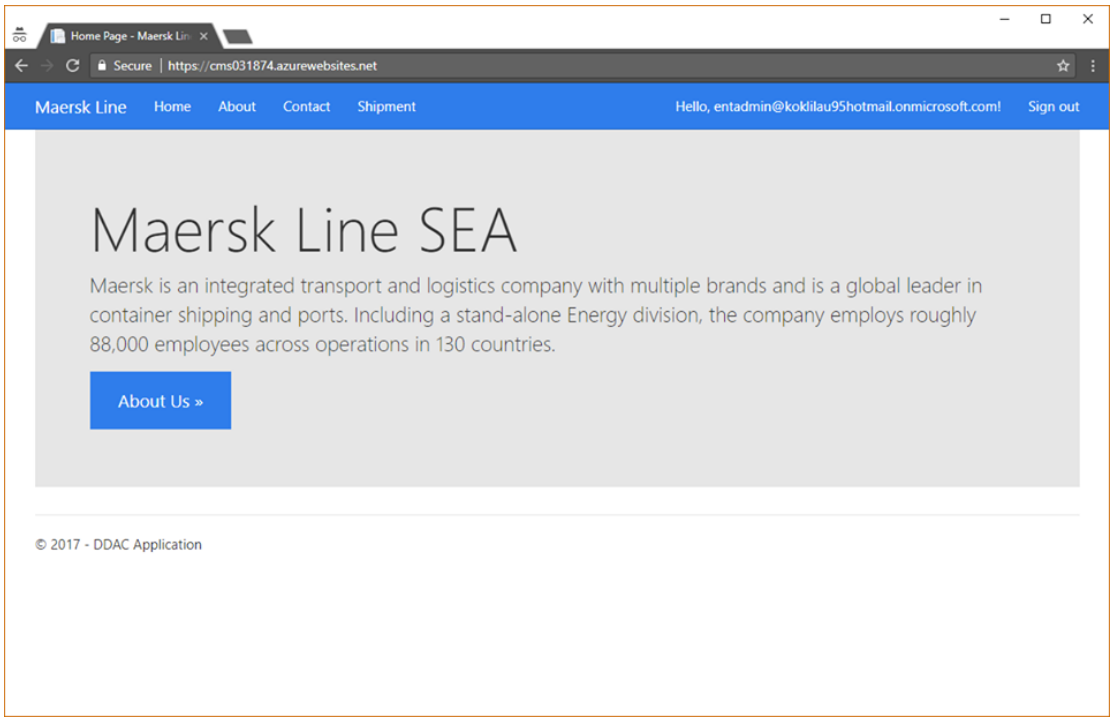
- In visual studio, publish the project and choose "Import" option to get the downloaded publish profile.



- After imported the selected publish profile, all the configuration will be set automatically and ready to be published.

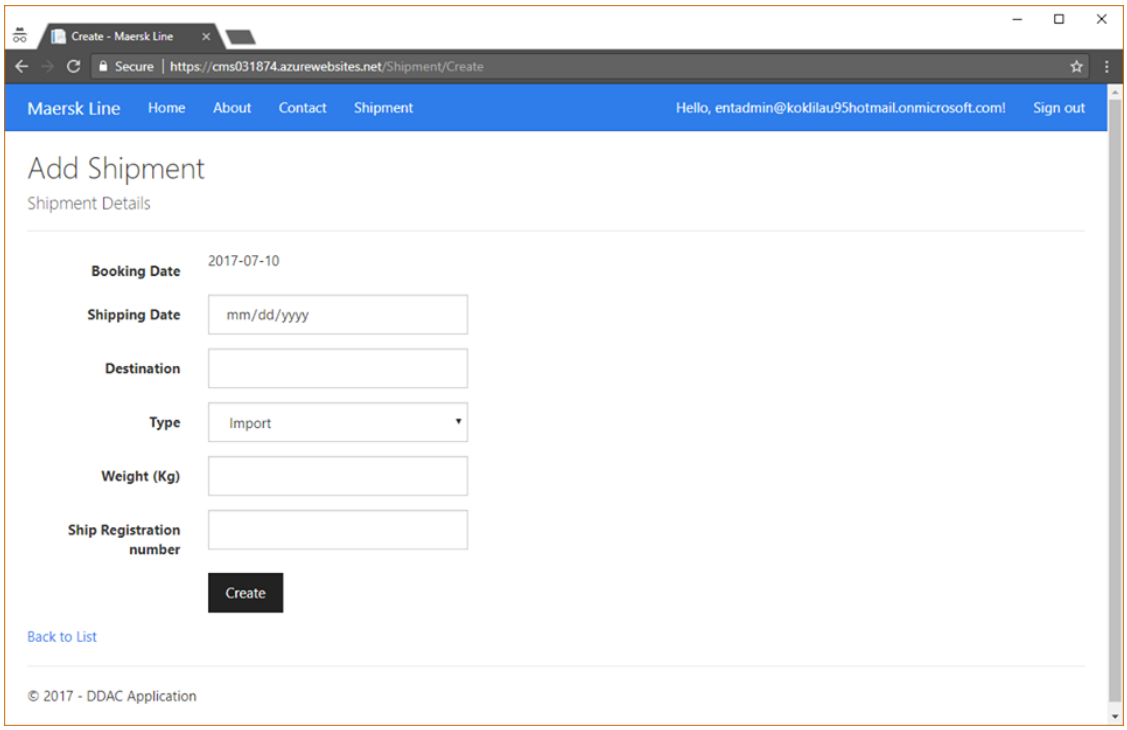


- Lastly, the web application is successfully deployed in Southeast Asia region.

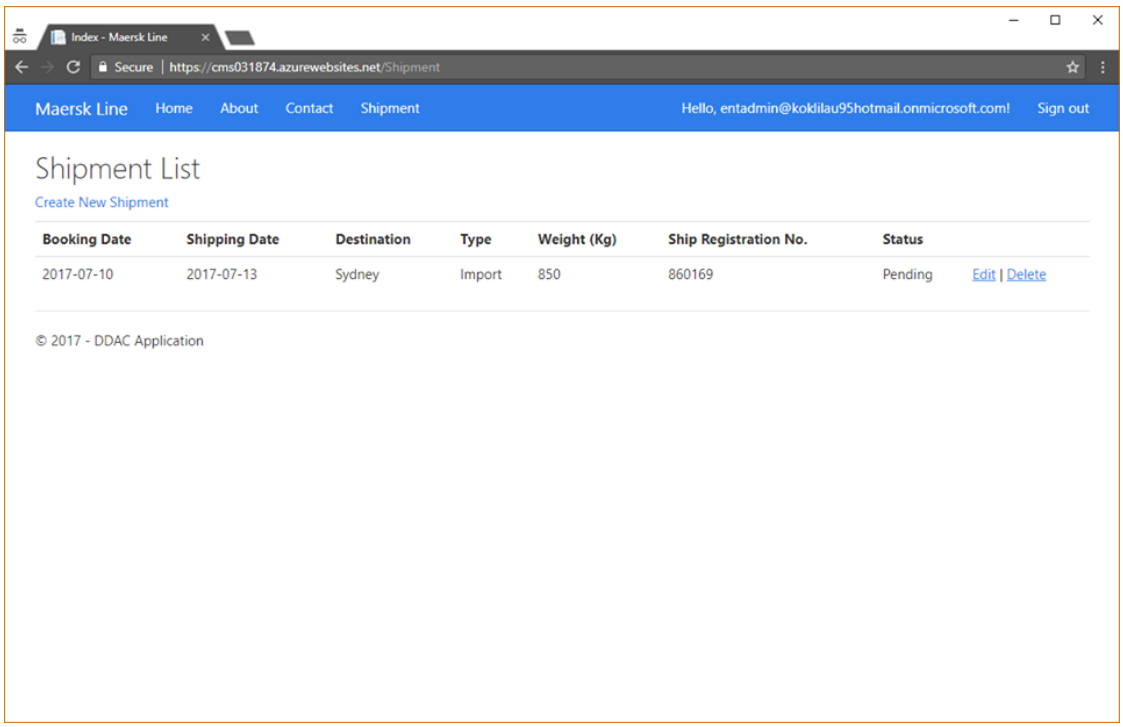


Web Application Pages

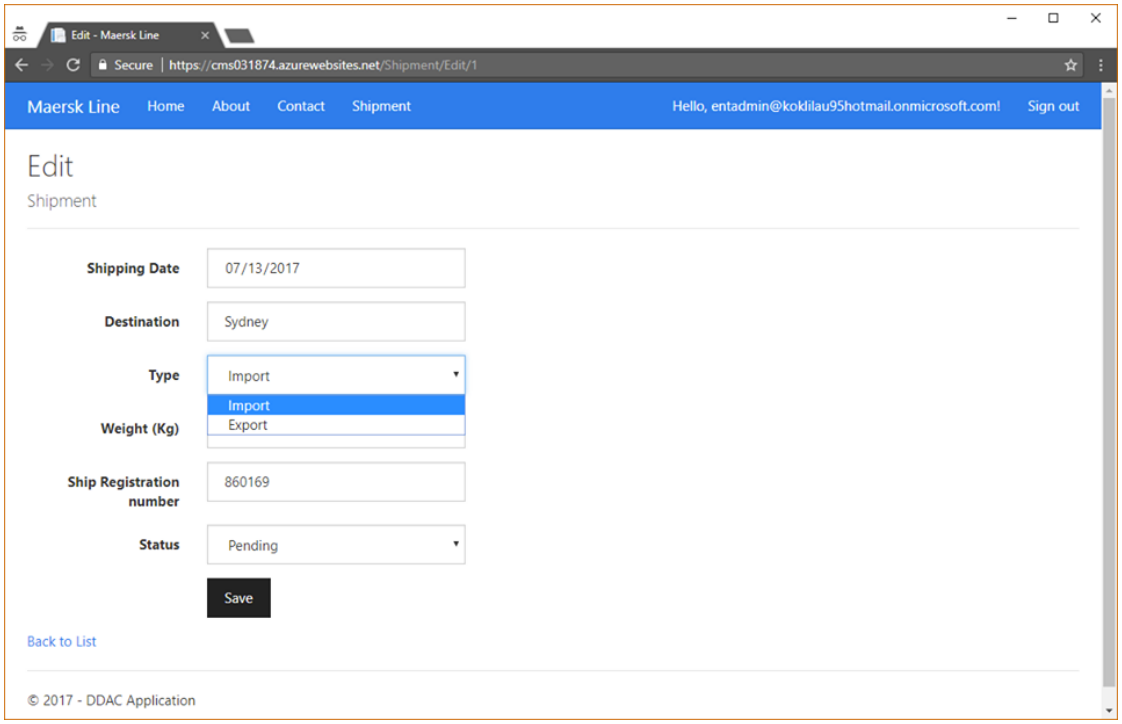
- This is the create shipment page for user to enter the shipment details.



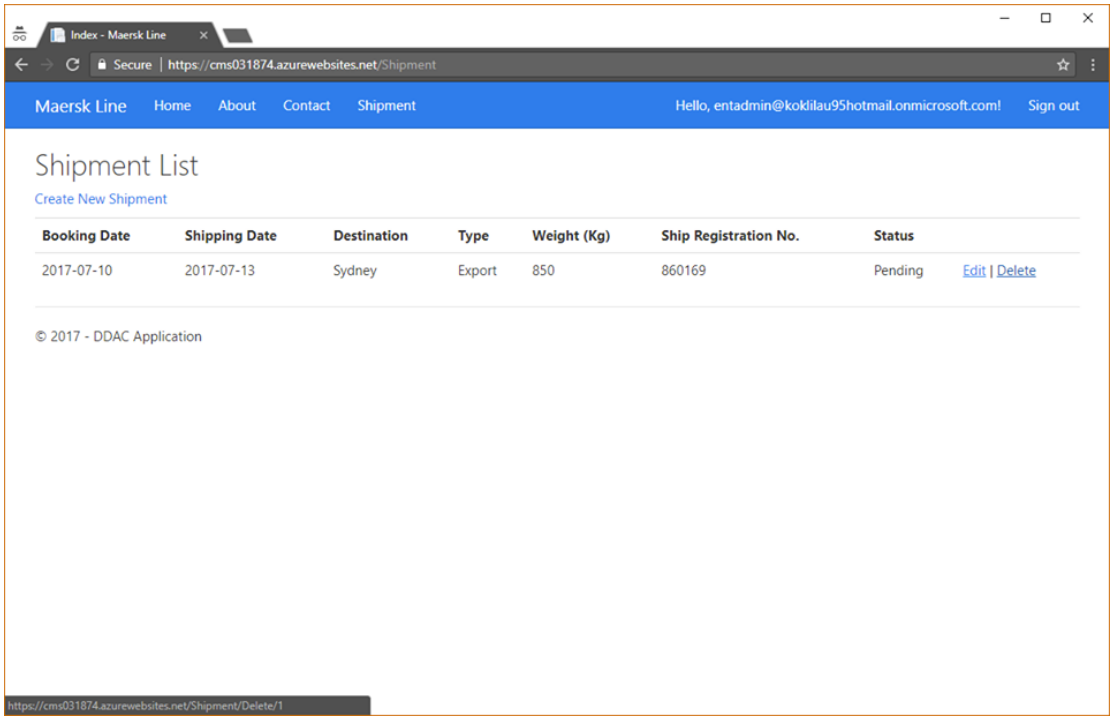
- This is the shipment list which show all the shipment created.



- This is the edit shipment page which allow user to make updates on selected shipment.

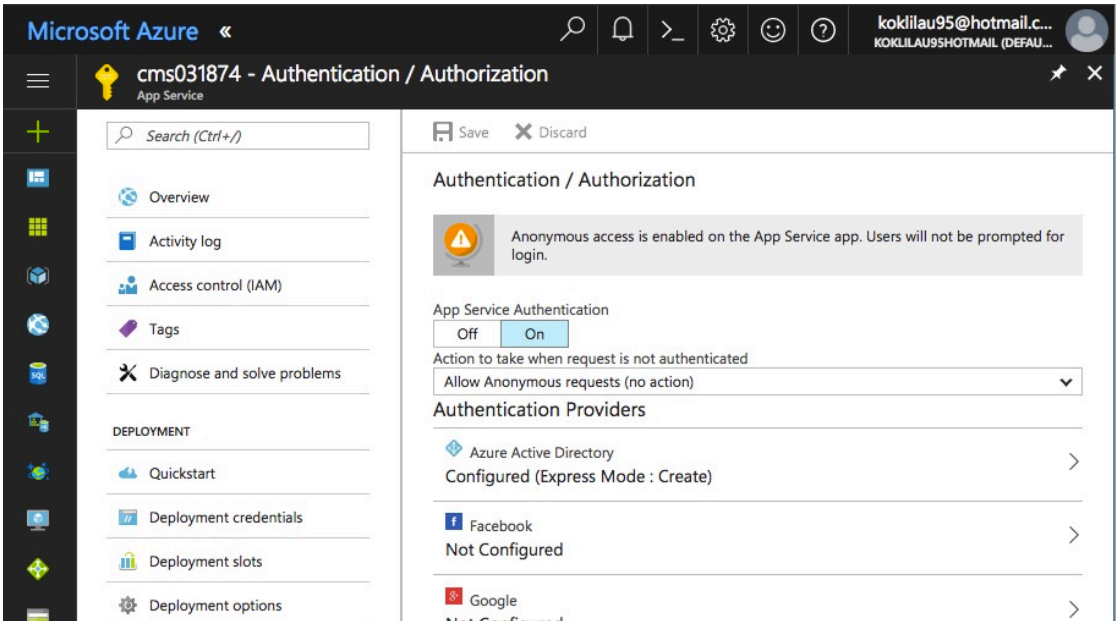


- After changes has been saved, the list will be refresh to the latest updates of shipment details



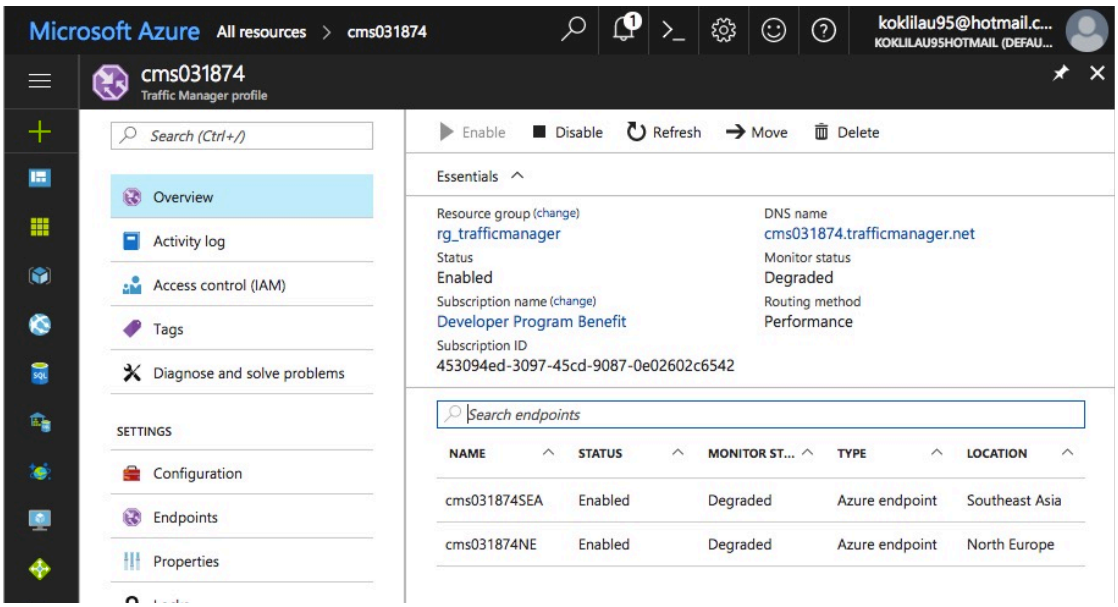
Configure secure access with Azure Active Directory

- In Azure portal, select “Authentication / Authorization” option and turn on the App Service Authentication.
- Then, configure the Azure Active Directory under authentication providers and make “Express” in management mode for automation.



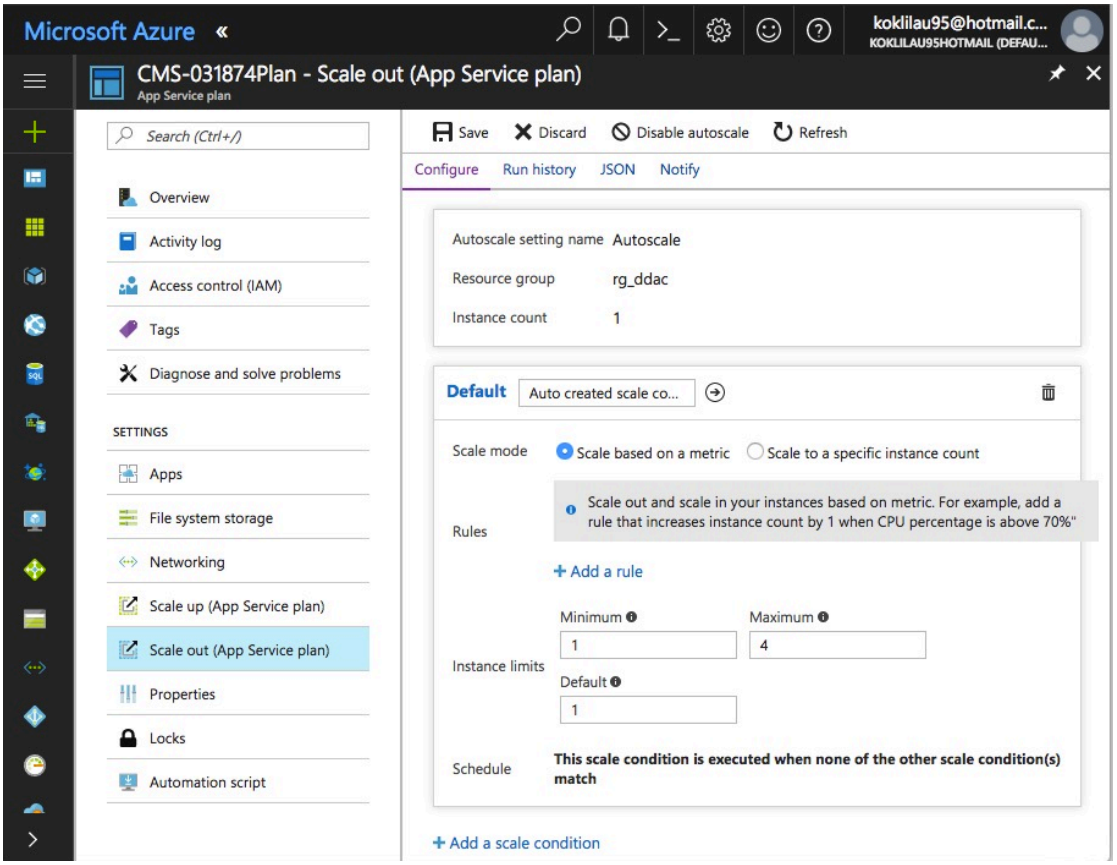
Configure Traffic Manager

- In Azure portal, create traffic manager profile within new resource group (rg_trafficmanager).
- Two endpoints is created for Southeast Asia region web application as well as North Europe region.



Enhance Performance with Application Scaling

- In Azure portal, select “Scale out” option in App Service plan in order to turn on “Autoscale” feature. In this configuration, maximum of 4 instances has set to be scale automatically to improve the application scalability.



Test Plan and Testing Discussion

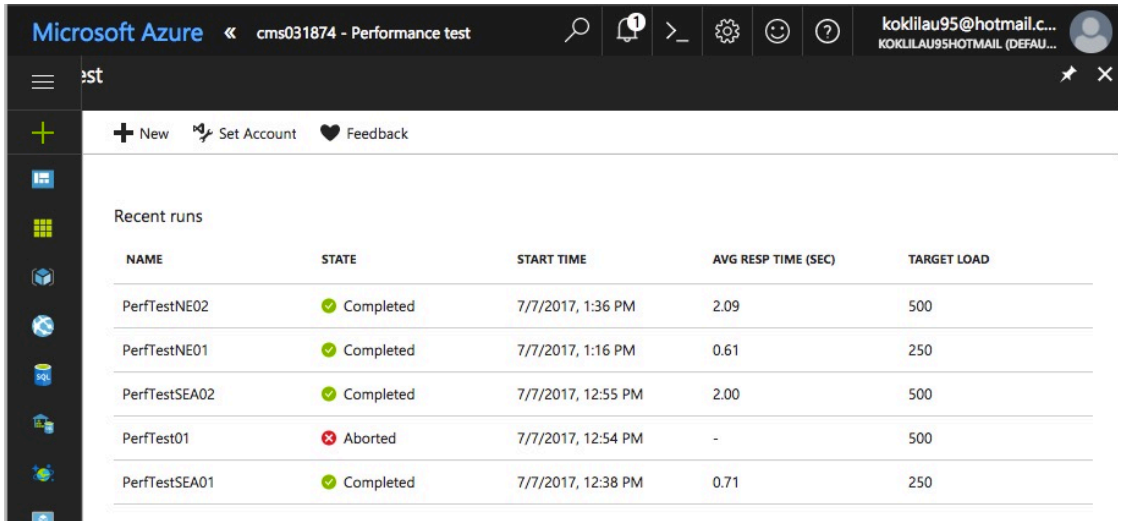
Test Plan

Test Case	Test Descriptions	Expected Result	Actual Result	Status	Remarks
A. Web Application					
Test01	Browse web application and sign in with authorized email account.	Redirect to the system page with logged in account.	As expected	Pass	
Test02	Create new shipment by filling the details and then click on “create” button.	New shipment has been created.	As expected.	Pass	
Test03	Click on the “Edit” button beside the record and make changes of the shipment details.	Shipment details have been updated.	As expected.	Pass	
Test04	Sign out the system page.	User has signed out.	As expected	Pass	

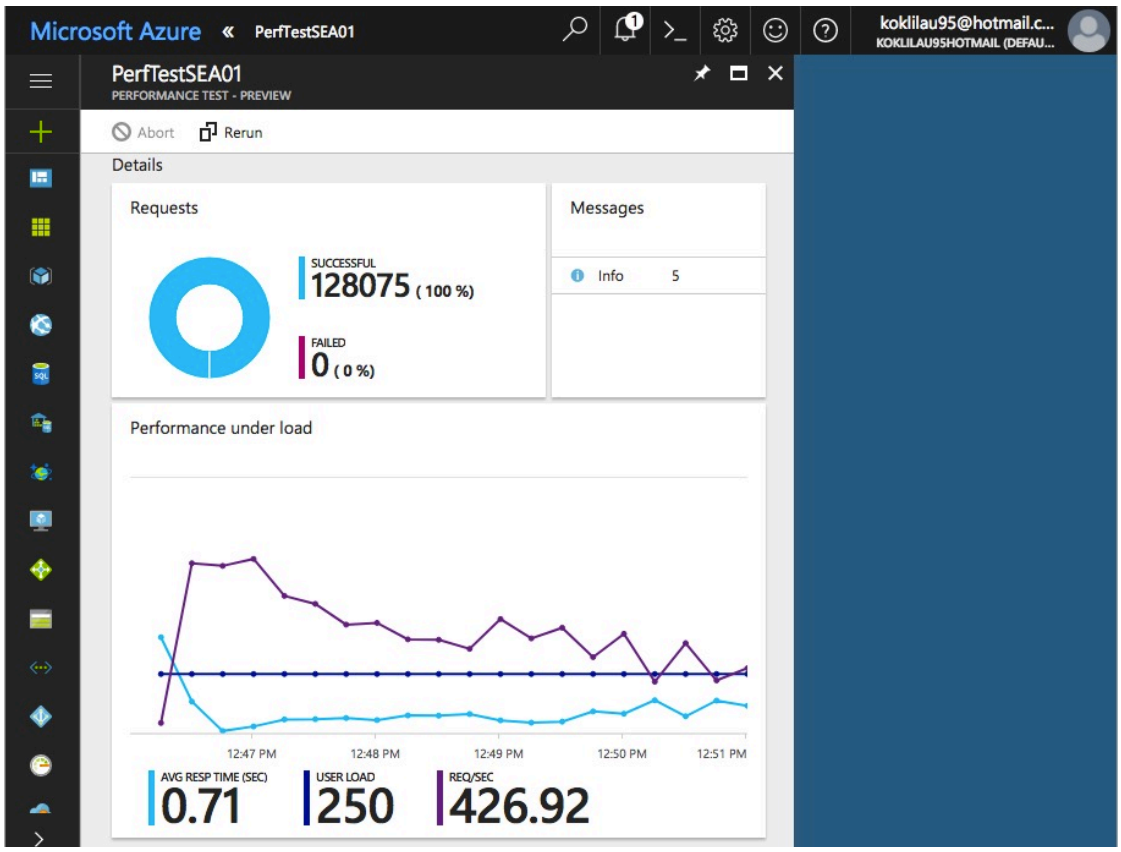
Test Case	Test Descriptions	Expected Result	Actual Result	Status	Remarks
B. Azure					
Test05	Published the web application to Azure Southeast Asia region.	Web application has been published.	As expected.	Pass	
Test06	Published the web application to Azure West Europe region.	Web application has been published.	As expected.	Pass	
Test07	Stop Southeast Asia region web application and run browse with Traffic Manager profile.	Auto routed to web application in North Europe region.	As expected.	Pass	

Web Application Performance Analysis

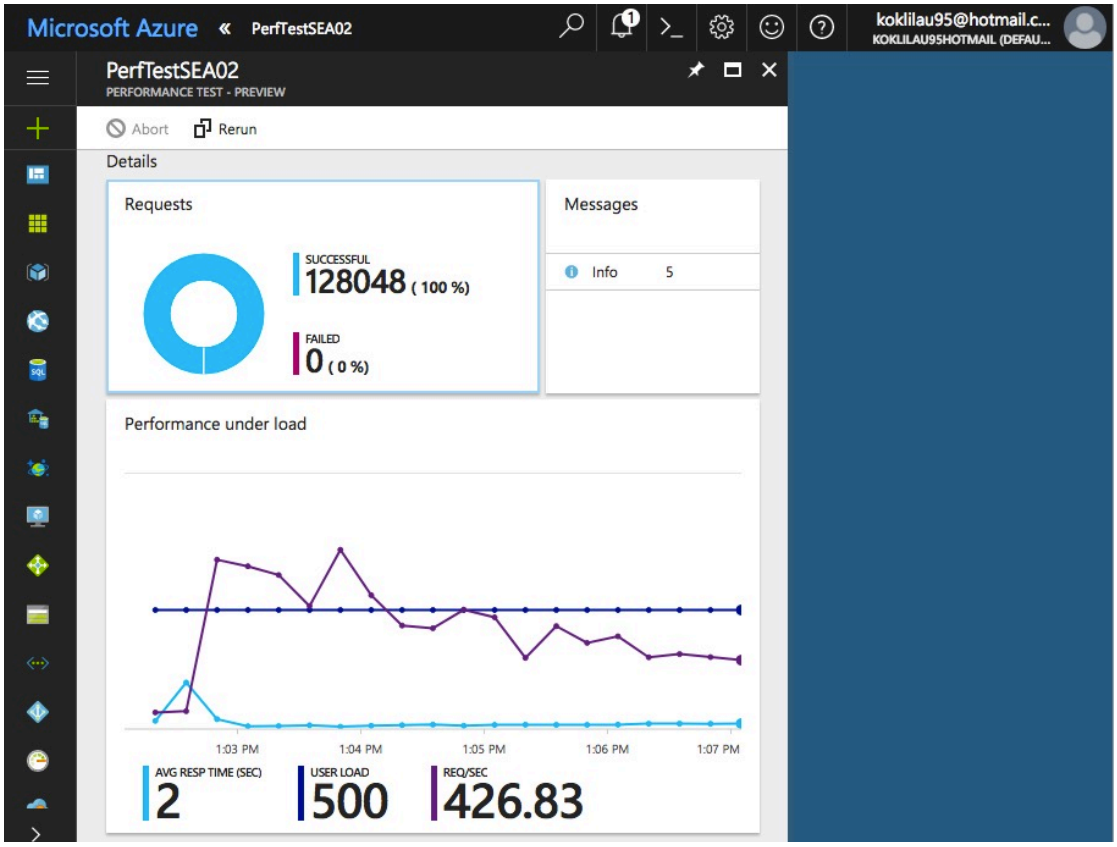
- Following is the list of performance testing completed.



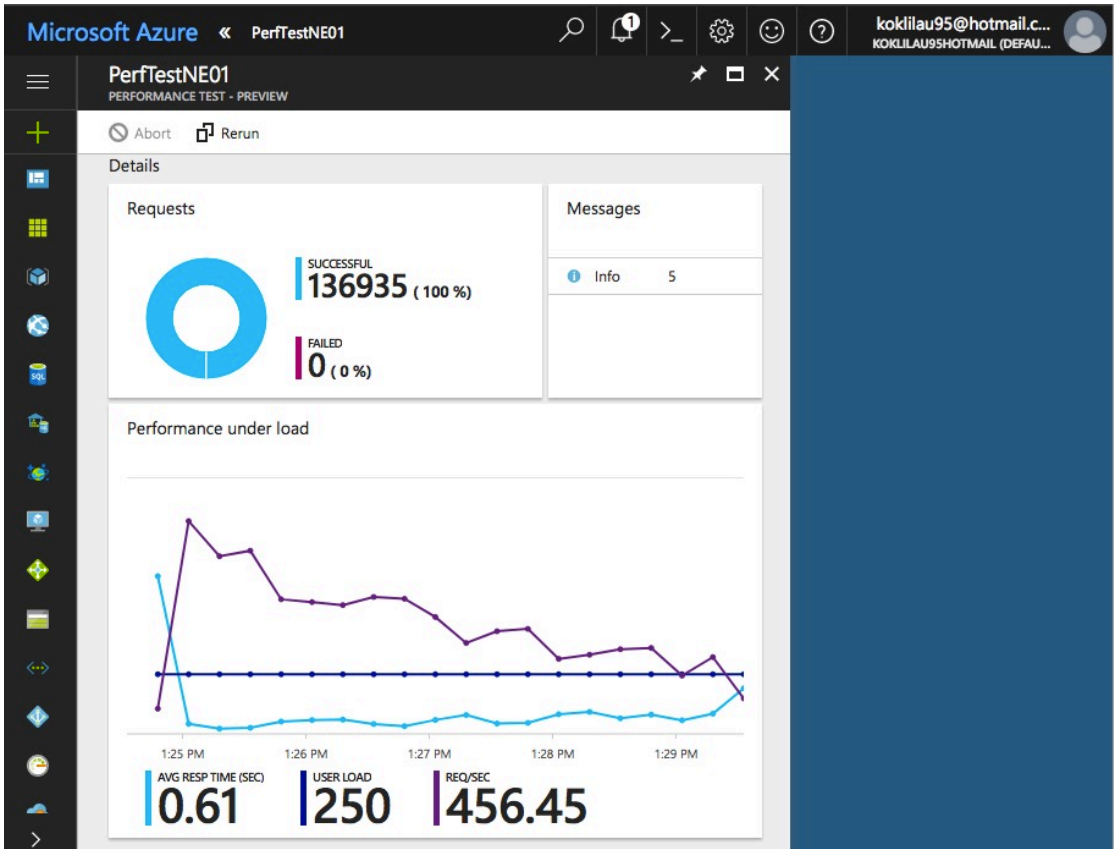
- Performance testing on Southeast Asia web application
    - 250 concurrent users testing (from SEA)



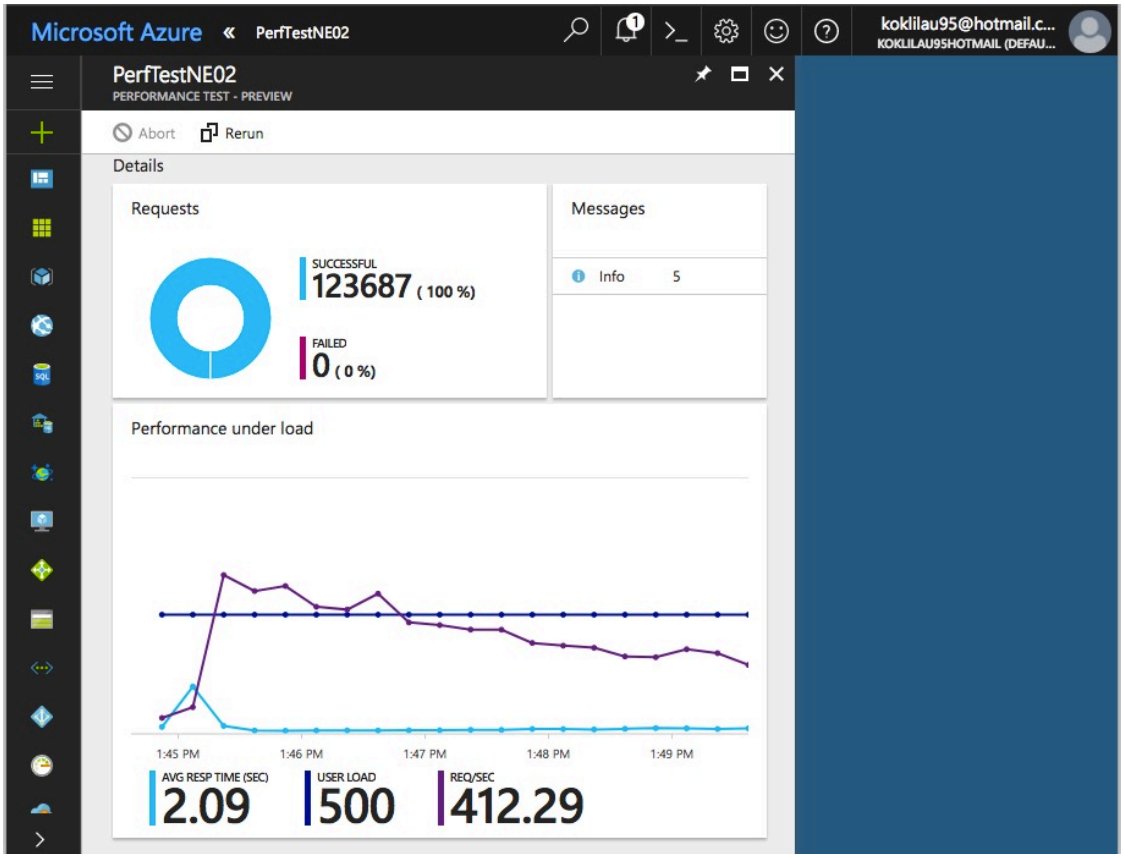
- 500 concurrent users testing (from SEA)



- 250 concurrent users testing (from NE)



- 500 concurrent users testing (from NE)

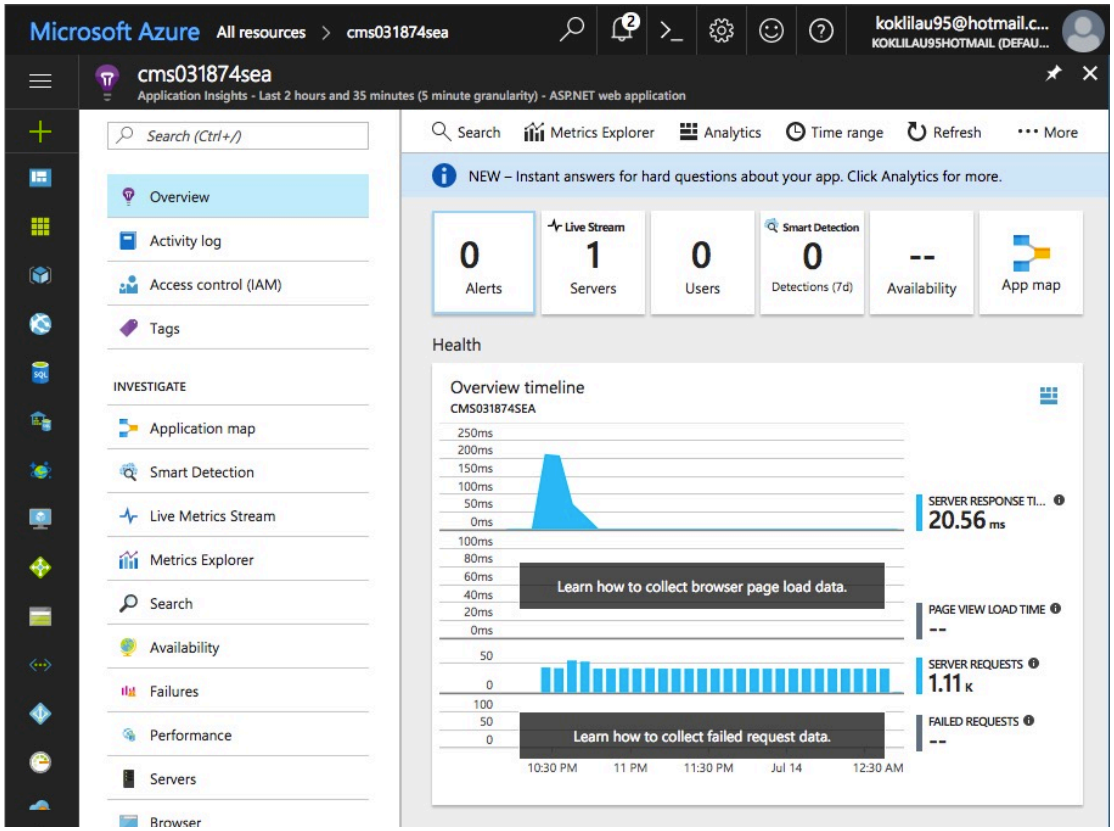


Performance Test	PerfTestSEA01	PerfTestSEA02	PerfTestNE01	PerfTestNE02
User Load	250	500	250	500
Avg. Resp Time (Sec)	0.71	2	0.61	2.09
Testing Time	5 minutes	5 minutes	5 minutes	5 minutes
Request/sec	426.92	426.83	456.45	412.29
Successful Requests	128075 (100%)	128048 (100%)	136935 (100%)	123687 (100%)
Failed Requests	0 (0%)	0 (0%)	0 (0%)	0 (0%)
CPU Time	1	1	1	1
Memory Working Set	5.12 B	5.13 B	5.13 B	5.13 B

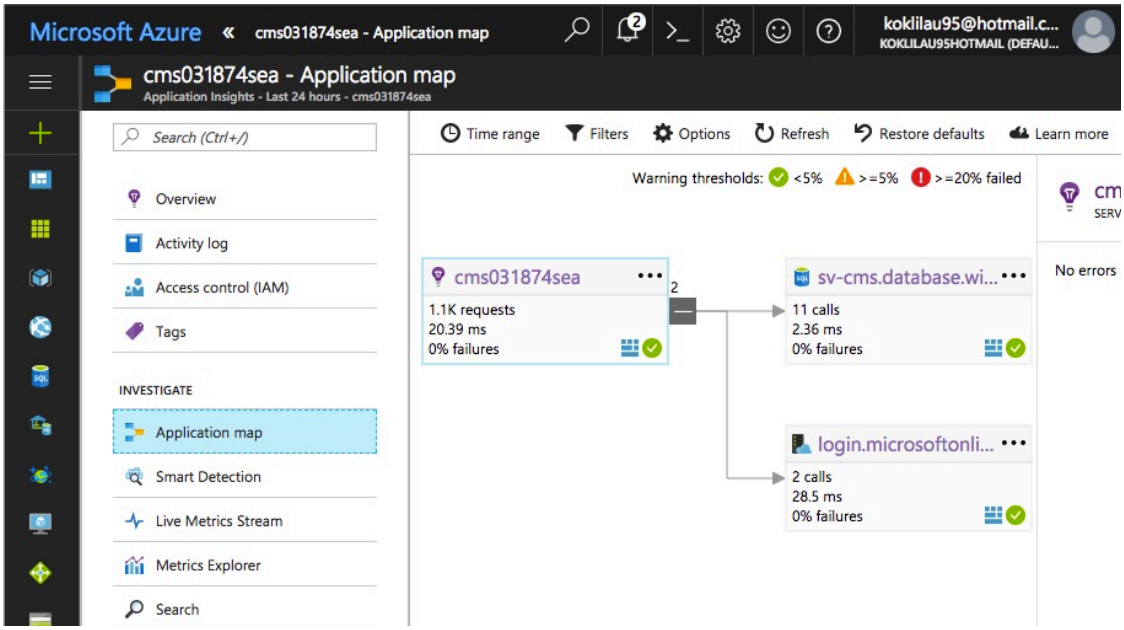
Based on these testing performed, the average response time is slightly higher which have more user load in the same time. The response time can be reduced with “AutoScale” feature from the scale out of App Service plan to improve its scalability.

Health Analysis with Application Insights

- Overall Health
    - Following is the health analysis of overall timeline.



- Application map
    - This is application map of web application which has linkage between active directory and SQL server. All the elements within application map has good connection with no errors.



Conclusion

Throughout the entire system development and deployment process, I have learnt the benefits of using Microsoft Azure cloud services for hosting website and cloud database along with server. I was able to publish web application using Visual Studio 2015 which synchronizing to Microsoft Azure during the deployment process.

Microsoft Azure gave developer the advantage of handling the heavy load traffic using traffic manager feature. It improved the performance of published web applications while able to redirect the traffic according to its selected regions. Moreover, Microsoft Azure able to monitor the performance of web application as well as other services which allow developer to analyse the required quality of services and choose the most suitable service plan in order to increase business profits.

Lastly, there are a lot features and services yet to be implemented for this project but Microsoft Azure is definitely a good cloud services provider which gave developer an opportunity to manage and enhance the performance of hosted web application.

Reference

WaveMaker. (2017). Rapid Application Development Model. [online] Available at: https://www.wavemaker.com/rapid-application-development-model/ [Accessed 7 Jun. 2017].