CT071-3-3-DDAC

INDIVIDUAL ASSIGNMENT

Name	**:**	NG SENG YONG
Student ID	**:**	TP034179
Intake Code	**:**	UC3F1706SE
Module Code	**:**	CT071-3-3-DDAC
Lecturer Name	**:**	DR. KALAI ANAND A/L RATNAM
Hand in Date	**:**	**13th APRIL 2018**

Acknowledgement

Firstly, developer would like to thank DR. Kalai for giving him an opportunity to be the complete the Maersk Line assignment. First of all, the developer would like to thank everyone who took part in the development of this project especially DR. Kalai. Without the kind supports and helps from him and other sources, the development of this project could not be successfully completed on time, especially the knowledge for Azure Cloud Platform.

Moreover, the developer would also like to show his appreciation towards his lecturer again, DR. Kalai. DR. Kalai has spared a lot of his precious time for guiding the developer to finish the development of the project. DR. Kalai has also given some references and also list out the mistakes that the developer is unaware in his development progression of Maersk Line and ensure the quality of this solution has met the requirements from him.

Lastly, developer would like to thank the course mate in guiding and helping him in the knowledge of C# and ASP.NET with the MVC framework, else, developer would not able to complete the project on time.

Table of Contents

1.0 INTRODUCTION

1.1 Project Background

Maersk Line is the largest container shipping company in the world that have customers through 374 offices in 116 countries all around the world and it was founded in 1928. Besides, it is the global container division and the largest operating unit of the A.P. Moller – Maersk Group, a Danish business conglomerate and has employed approximately 7,000 sea farers and 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.

In order to increase organizational flexibility and support further business growth of Maersk Line, the company has decided to consolidate all of its data centres and server rooms operating worldwide onto a virtualized platform which is Microsoft Azure. Some of the Maersk's IT environment has already hosted in Microsoft Azure and Maersk Line are planning to expand the scope of the relationship with Microsoft for changing over its IT setup based on Microsoft Azure, starting with the desktop environment up to container management.

1.2 Objective

The objective of Maersk Line's project is reducing overall supply chain costs and an efficient way to manage logistics by designing and developing a cloud based Container Management System (CMS) to enhance the business process flow.

1.3 Scope

The scope of this project is to design and develop a cloud based CMS to cater to manage the containers. The cloud based CMS is hosting in Microsoft Azure platform which will allow users to access it whenever and wherever they want. There are some criteria to fulfil:

1. From import, export and transhipment processing to gate operations.
2. To be able to scale the solution to meet the needs of demands during peak seasons.

3. Improves profitability, reduce 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 haulier vehicles, delivering an exceptional level of automation and removing human error.
6. Manage your entire booking process from schedule search to booking confirmation

1.4 Project Specification

1. Scalability
   - To ensure that the CMS has the capacity to grow with the data when there are more users are using it.
2. Usability
   - To ensure that the system is easy to use, efficient, easy to learn, and satisfied by users.
3. Monitoring
   - To ensure that the processes of the application are monitored and it can identify the defects and errors in the system before it has highly impacted to the users.
4. Maintainability
   - To ensure that the maintenance is able to repair or troubleshoot or update the system within a given time.
5. Availability
   - To ensure that the system is always available when the users are required to use the system.

1.5 Project Deliverables

There is a summary of the major functions or contents of the CMS's project which is shown as below.

User	Project Deliverables
Admin	- Login - Register new agent - Create, edit, delete, view ship - Create, edit, delete, view customer - Create, edit, delete, view schedule - Create & view booking
Agent	- Login - Create & view booking - Create, edit, delete, view customer

2.0 Project Plan

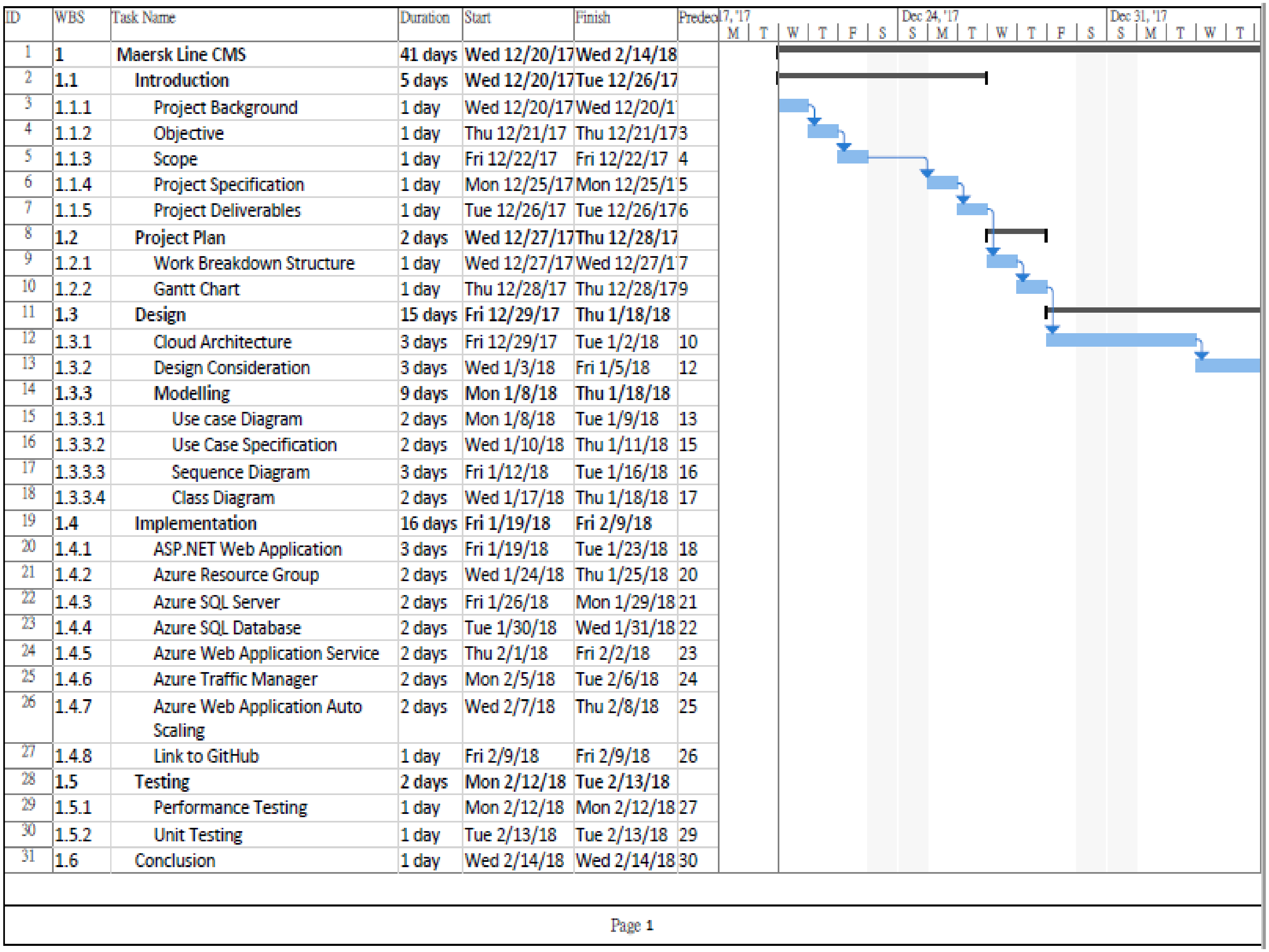
2.1 Work Breakdown Structure (WBS)

WBS	Task Name	Duration
1	**Maersk Line CMS**	**41 days**
1.1	**Introduction**	**5 days**
1.1.1	Project Background	1 day
1.1.2	Objective	1 day
1.1.3	Scope	1 day
1.1.4	Project Specification	1 day
1.1.5	Project Deliverables	1 day
1.2	**Project Plan**	**2 days**
1.2.1	Work Breakdown Structure	1 day
1.2.2	Gantt Chart	1 day
1.3	**Design**	**15 days**
1.3.1	Cloud Architecture	3 days
1.3.2	Design Consideration	3 days
1.3.3	**Modelling**	**9 days**
1.3.3.1	Use case Diagram	2 days
1.3.3.2	Use Case Specification	2 days
1.3.3.3	Sequence Diagram	3 days
1.3.3.4	Class Diagram	2 days

1.4	**Implementation**	**16 days**
1.4.1	ASP.NET Web Application	3 days
1.4.2	Azure Resource Group	2 days
1.4.3	Azure SQL Server	2 days
1.4.4	Azure SQL Database	2 days
1.4.5	Azure Web Application Service	2 days
1.4.6	Azure Traffic Manager	2 days
1.4.7	Azure Web Application Auto Scaling	2 days
1.4.8	Link to GitHub	1 day
1.5	**Testing**	**2 days**
1.5.1	Performance Testing	1 day
1.5.2	Unit Testing	1 day
1.6	Conclusion	1 day

Table 1: Work Breakdown Structure

2.2 Gantt Chart



ID	WBS	Task Name	Duration	Start	Finish	Predec
1	1	Maersk Line CMS	41 days	Wed 12/20/17	Wed 2/14/18	
2	1.1	Introduction	5 days	Wed 12/20/17	Tue 12/26/17	
3	1.1.1	Project Background	1 day	Wed 12/20/17	Wed 12/20/1	
4	1.1.2	Objective	1 day	Thu 12/21/17	Thu 12/21/17	3
5	1.1.3	Scope	1 day	Fri 12/22/17	Fri 12/22/17	4
6	1.1.4	Project Specification	1 day	Mon 12/25/17	Mon 12/25/1	5
7	1.1.5	Project Deliverables	1 day	Tue 12/26/17	Tue 12/26/17	6
8	1.2	Project Plan	2 days	Wed 12/27/17	Thu 12/28/17	
9	1.2.1	Work Breakdown Structure	1 day	Wed 12/27/17	Wed 12/27/1	7
10	1.2.2	Gantt Chart	1 day	Thu 12/28/17	Thu 12/28/17	9
11	1.3	Design	15 days	Fri 12/29/17	Thu 1/18/18	
12	1.3.1	Cloud Architecture	3 days	Fri 12/29/17	Tue 1/2/18	10
13	1.3.2	Design Consideration	3 days	Wed 1/3/18	Fri 1/5/18	12
14	1.3.3	Modelling	9 days	Mon 1/8/18	Thu 1/18/18	
15	1.3.3.1	Use case Diagram	2 days	Mon 1/8/18	Tue 1/9/18	13
16	1.3.3.2	Use Case Specification	2 days	Wed 1/10/18	Thu 1/11/18	15
17	1.3.3.3	Sequence Diagram	3 days	Fri 1/12/18	Tue 1/16/18	16
18	1.3.3.4	Class Diagram	2 days	Wed 1/17/18	Thu 1/18/18	17
19	1.4	Implementation	16 days	Fri 1/19/18	Fri 2/9/18	
20	1.4.1	ASP.NET Web Application	3 days	Fri 1/19/18	Tue 1/23/18	18
21	1.4.2	Azure Resource Group	2 days	Wed 1/24/18	Thu 1/25/18	20
22	1.4.3	Azure SQL Server	2 days	Fri 1/26/18	Mon 1/29/18	21
23	1.4.4	Azure SQL Database	2 days	Tue 1/30/18	Wed 1/31/18	22
24	1.4.5	Azure Web Application Service	2 days	Thu 2/1/18	Fri 2/2/18	23
25	1.4.6	Azure Traffic Manager	2 days	Mon 2/5/18	Tue 2/6/18	24
26	1.4.7	Azure Web Application Auto Scaling	2 days	Wed 2/7/18	Thu 2/8/18	25
27	1.4.8	Link to GitHub	1 day	Fri 2/9/18	Fri 2/9/18	26
28	1.5	Testing	2 days	Mon 2/12/18	Tue 2/13/18	
29	1.5.1	Performance Testing	1 day	Mon 2/12/18	Mon 2/12/18	27
30	1.5.2	Unit Testing	1 day	Tue 2/13/18	Tue 2/13/18	29
31	1.6	Conclusion	1 day	Wed 2/14/18	Wed 2/14/18	30

Figure 1: Gantt Chart

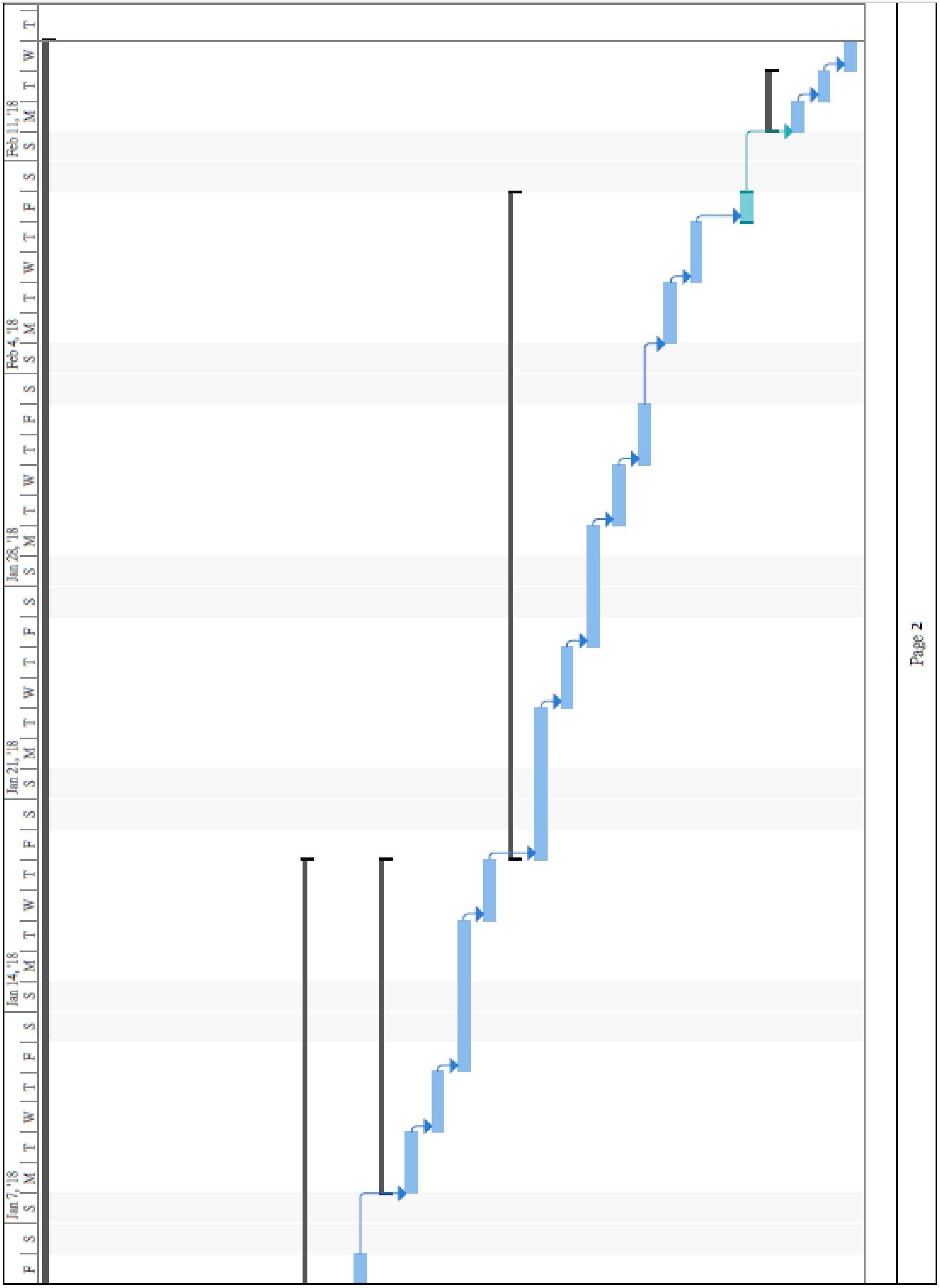


Figure 2: Gantt Chart

3.0 Design

3.1 Cloud Architecture Diagram

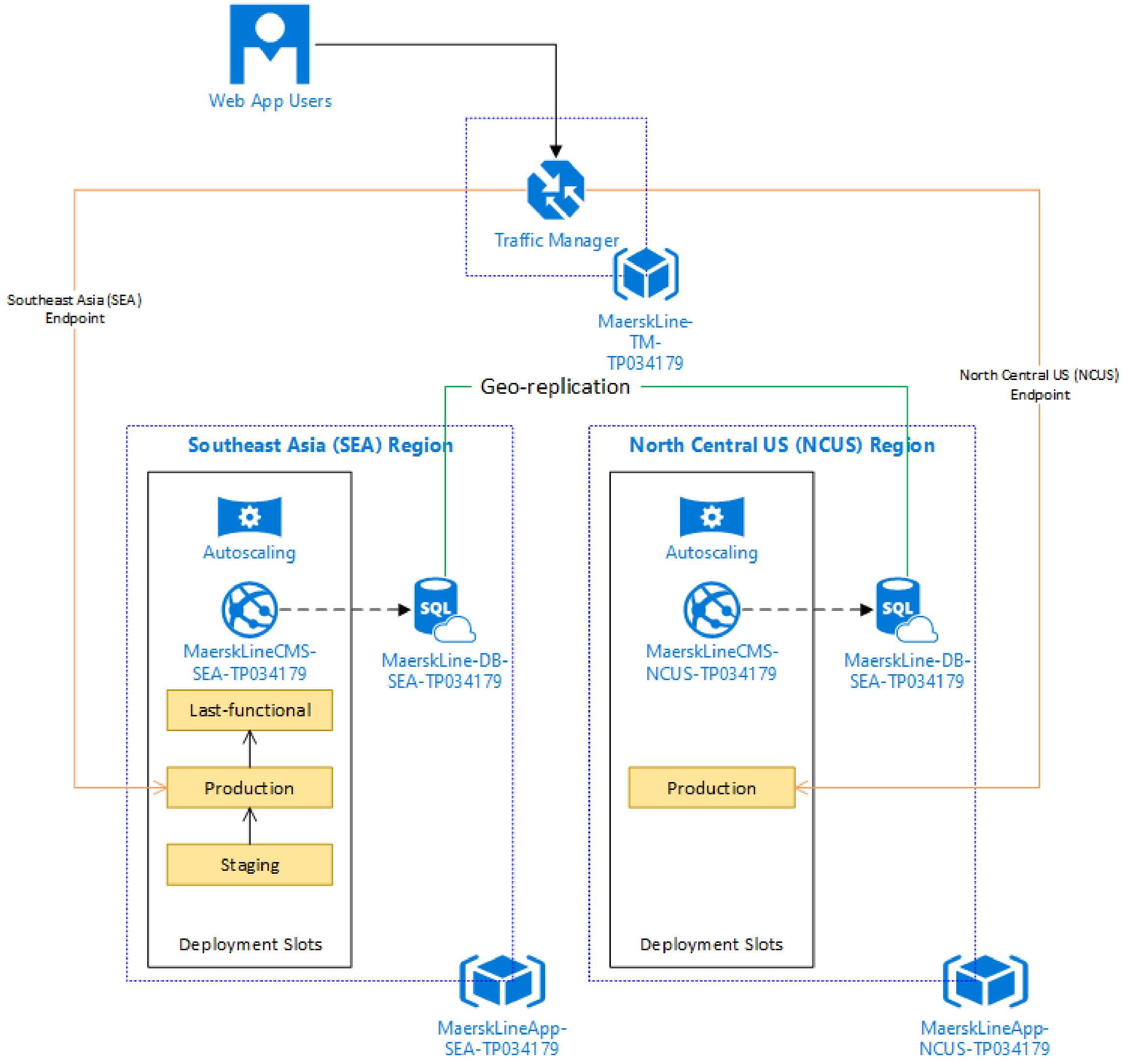

Web App Users
Traffic Manager
MaerskLine-TM-TP034179
Southeast Asia (SEA) Endpoint
North Central US (NCUS) Endpoint
Geo-replication
Southeast Asia (SEA) Region
North Central US (NCUS) Region
Autoscaling
MaerskLineCMS-SEA-TP034179
MaerskLine-DB-SEA-TP034179
Last-functional
Production
Staging
Deployment Slots
Autoscaling
MaerskLineCMS-NCUS-TP034179
MaerskLine-DB-SEA-TP034179
Production
Deployment Slots
MaerskLineApp-SEA-TP034179
MaerskLineApp-NCUS-TP034179

Figure 3: Cloud Architecture Diagram for Maersk Line

In Figure 3, it is the Cloud Architecture Diagram for Maersk Line TP034179. The diagram is showing how actually Maersk Line TP034179 works on the Azure Cloud Platform. This diagram has two regions that are Southeast Asia (SEA) region and North Central US (NCUS) region; where SEA region is the primary cloud service and NCUS is the secondary cloud service for the Maersk Line.

There will be 1 to 3 instances for each web app in a region for auto scaling services that is setup by the developer in Section 4.7. There are two endpoints for the SEA and

NCUS region's Web App. The endpoints are used for managing the performance by the traffic manager as shown in the figure above. The SQL database for SEA region is MaerskLine-DB-SEA-TP034179, where NCUS region's SQL database is also MaerskLine-DB-SEA-TP034179. It is because the database of the NCUS region is the secondary database that replicated from the primary database of SEA region. Therefore, there are sharing the same SQL database but different SQL server. The geo-replication is implemented as mentioned earlier for enabling secondary database for NCUS region. It can also be used for failover when the primary SQL database is broken or failed, the secondary database at NCUS region will replace as the primary database at that moment. In this case, Maersk Line SQL database will always be available for the users to access.

3.2 Design Consideration

There are several assumptions or considerations before developing and designing the Maersk Line web application. First assumption is Maersk Line will not be only available in Southeast Asia (SEA) region, there will be also available in North Central US (NCUS) region. However, there will only be available in these two regions as there are insufficient credits for hosting the web application in Azure for more than two regions. Besides, the performance testing will only conduct for SEA region's web application service. Lastly, the traffic manager's region will locate at SEA region.

3.3 Modelling

3.3.1 Use Case Diagram

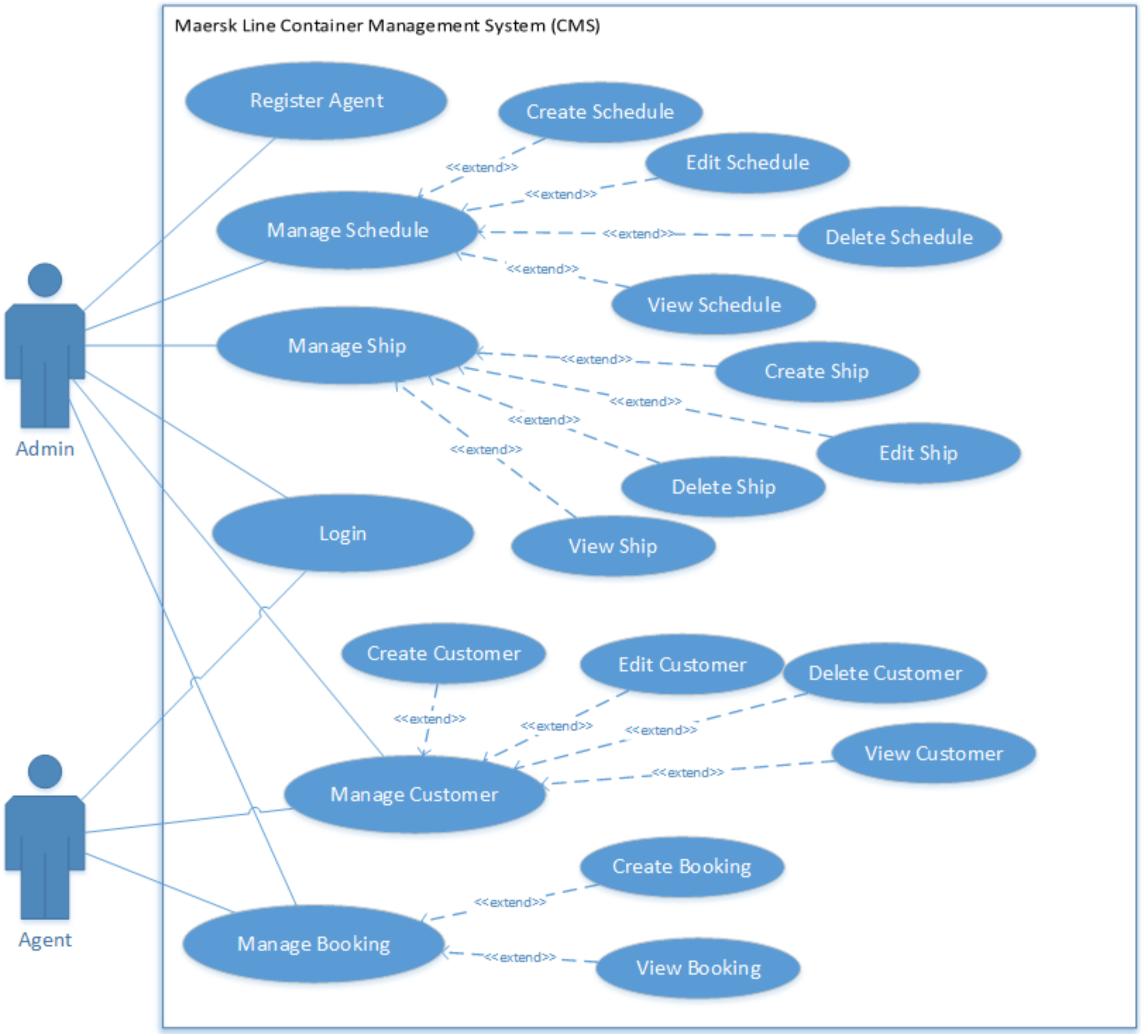


Figure 4: Use Case Diagram

3.3.2 Use Case Specification

Use Case Name:	**Login**
Summary:	Admin and agent login to the system for accessing more functions.
Dependency:	N/A
Actor(s):	Admin, Agent
Precondition(s):	Admin, Agent must have registered a username and password.
Description of main sequence:	1. Enter username and password. 2. Submit 3. System will verify the credentials. 4. Login successfully
Description of alternative sequence:	3(a). If username and password not match, error message will be prompted.
Post condition:	Admin, Agent are directed to their home page.

Table 2: Use case specification for Login

Use Case Name:	**Register Agent**
Summary:	Admin register agent account with agent’s email address, and password.
Dependency:	N/A
Actor(s):	Admin
Precondition(s):	Admin must login into the system.
Description of main sequence:	1. Enter agent’s email address, and password. 2. System will verify the credentials. 3. Register successfully.
Description of alternative sequence:	2(a). If email address, password is not matching the validation criteria, error message will be shown.

Post condition:	Agent account has successfully created by admin.

Table 3: Use Case Description for Register Agent

Use Case Name:	**Manage Schedule**
Summary:	Admin is able to create, edit, delete, and view schedule.
Dependency:	<<extend>> Create Schedule <<extend>> Edit Schedule <<extend>> Delete Schedule <<extend>> View Schedule
Actor(s):	Admin
Precondition(s):	Admin must login into the system.
Description of main sequence:	*Create Schedule* 1. Click “Create New Schedule” 2. Enter Schedule information 3. Submit by clicking “Create” 4. Schedule is created successfully. 5. View the created schedule’s information. *Edit Schedule* 1. Click “Edit” 2. Enter the new schedule information 3. Submit by clicking “Save” 4. Schedule is edited successfully. *Delete Schedule* 1. Click “Delete” 2. Confirm deletion by clicking “Delete” again 3. Schedule is deleted successfully.

	View Schedule 1. Click “Details” 2. View the schedule’s information
Description of alternative sequence:	*Create Schedule* 5(a). Add a new schedule, edit the current schedule, back to a list of schedules. *View Schedule* 2(a). Add a new schedule, edit the current schedule, back to a list of schedules. *Delete Schedule* 2(a). Fail to delete schedule due to one or more ships have been assigned to this schedule.
Post condition:	*Edit Schedule* Redirect back to the list of schedules. *Delete Schedule* Redirect back to the list of schedules.

Table 4: Use Case Description for Manage Schedule

Use Case Name:	**Manage Ship**
Summary:	Admin is able to create, edit, delete, and view ship.
Dependency:	<<extend>> Create Ship <<extend>> Edit Ship <<extend>> Delete Ship <<extend>> View Ship
Actor(s):	Admin

Precondition(s):	Admin must login into the system.
Description of main sequence:	*Create Ship* 1. Click “Create New Ship” 2. Enter Ship information 3. Submit by clicking “Create” 4. Ship is created successfully. 5. View the created ship’s information. *Edit Ship* 1. Click “Edit” 2. Enter the new ship information 3. Submit by clicking “Save” 4. Ship is edited successfully. *Delete Ship* 1. Click “Delete” 2. Confirm deletion by clicking “Delete” again 3. Ship is deleted successfully. *View Ship* 1. Click “Details” 2. View the ship’s information
Description of alternative sequence:	*Create Ship* 5(a). Edit the current ship, back to a list of ships. *View Ship* 2(a). Edit the current ship, back to a list of ships. *Delete Ship*

	2(a). Fail to delete ship due to one or more schedule have been assigned to this ship.
Post condition:	*Edit Ship* Redirect back to the list of ships. *Delete Ship* Redirect back to the list of ships.

Table 5: Use Case Description for Manage Ship

Use Case Name:	**Manage Customer**
Summary:	Admin is able to create, edit, delete, and view customer.
Dependency:	<<extend>> Create Customer <<extend>> Edit Customer <<extend>> Delete Customer <<extend>> View Customer
Actor(s):	Admin
Precondition(s):	Admin must login into the system.
Description of main sequence:	*Create Customer* 1. Click “Create New Customer” 2. Enter Customer information 3. Submit by clicking “Create” 4. Customer is created successfully. 5. View the created Customer’s information. *Edit Customer* 1. Click “Edit” 2. Enter the new Customer information 3. Submit by clicking “Save”

	4. Customer is edited successfully. *Delete Customer* 1. Click “Delete” 2. Confirm deletion by clicking “Delete” again 3. Customer is deleted successfully. *View Customer* 1. Click “Details” 2. View the Customer’s information
Description of alternative sequence:	*Create Customer* 5(a). Edit the current Customer, back to a list of Customers. *View Customer* 2(a). Edit the current Customer, back to a list of Customers.
Post condition:	*Edit Customer* Redirect back to the list of Customers. *Delete Customer* Redirect back to the list of Customers.

Table 6: Use Case Description for Manage Customer

Use Case Name:	**Manage Booking**
Summary:	Admin, agent is able to create and view booking.
Dependency:	<<extend>> Create Booking <<extend>> View Booking
Actor(s):	Admin, Agent
Precondition(s):	Admin, Agent must login into the system.

Description of main sequence:	*Create Booking* 1. Select “Booking” in the top navigation bar. 2. Select the desired schedule 3. Select the desired ship 4. Select customer 5. Enter the container information 6. Booking is created successfully *View Booking* 1. Select “View Booking” at top navigation bar 2. System has displayed a list of bookings
Description of alternative sequence:	3(a). If username and password not match, error message will be prompted.
Post condition:	*Create Booking* 6(a). Redirect user to View booking page

Table 7: Use Case Description for Manage Booking

3.3.3 Sequence Diagram

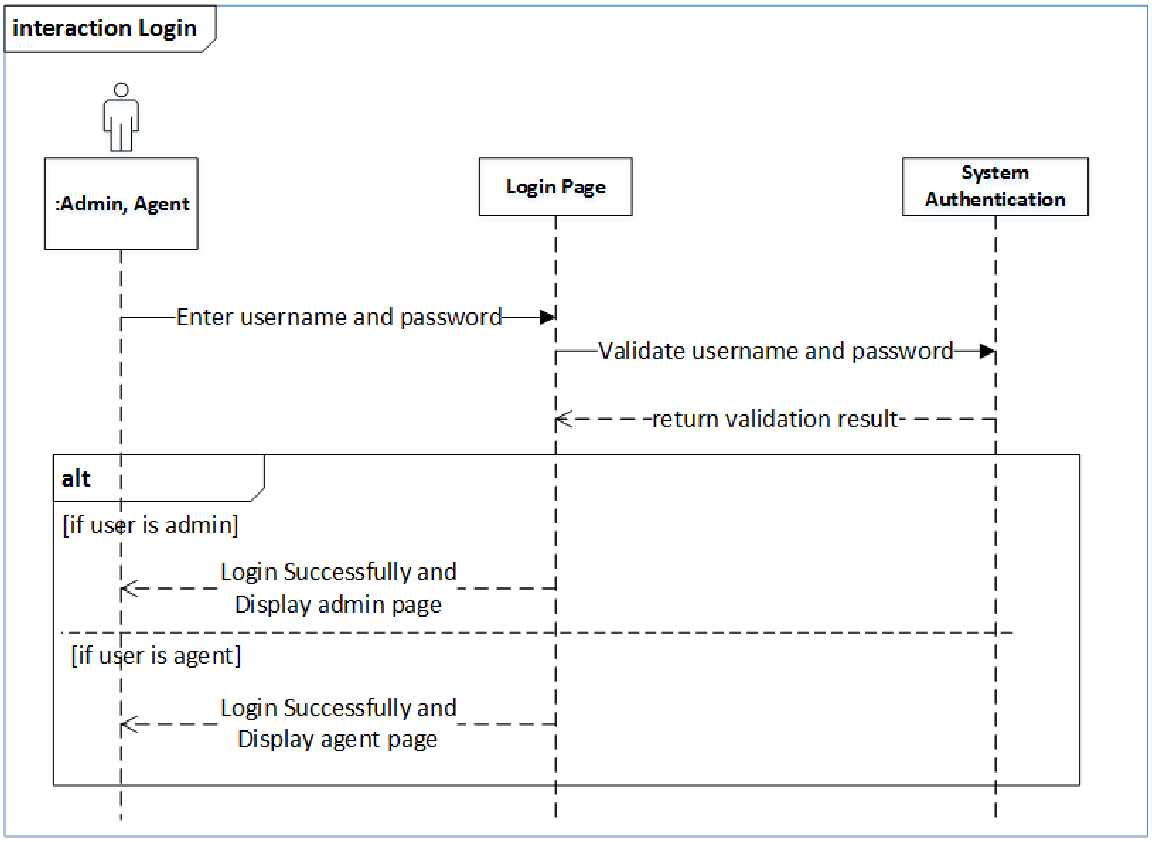

interaction Login
:Admin, Agent
Login Page
System Authentication
Enter username and password
Validate username and password
return validation result
alt
[if user is admin]
Login Successfully and Display admin page
[if user is agent]
Login Successfully and Display agent page

Figure 5: Login Sequence Diagram

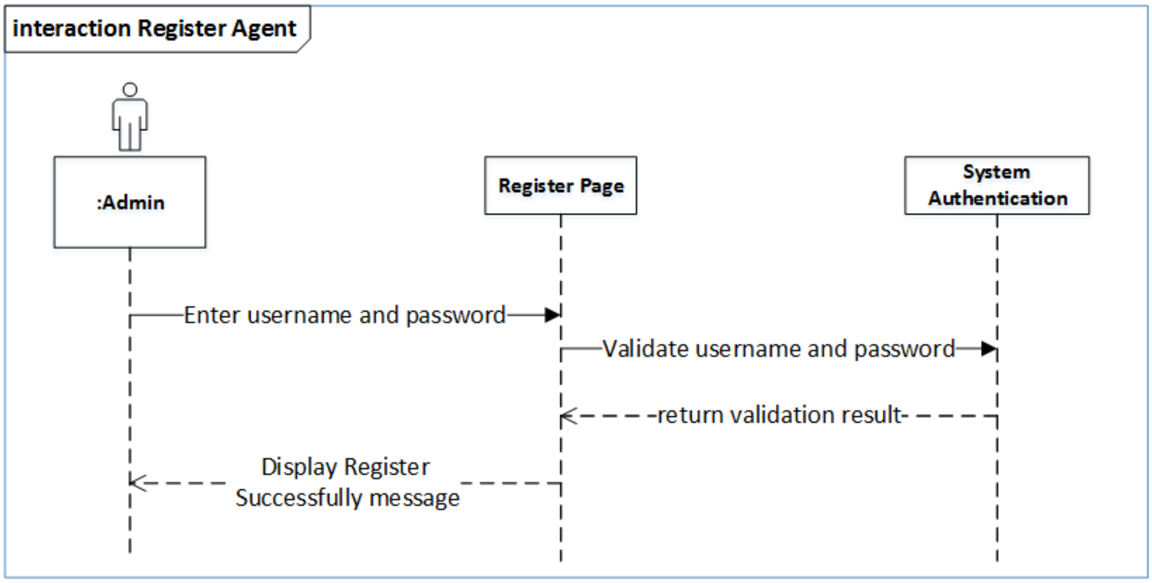

interaction Register Agent
:Admin
Register Page
System Authentication
Enter username and password
Validate username and password
return validation result
Display Register Successfully message

Figure 6: Register Agent Sequence Diagram

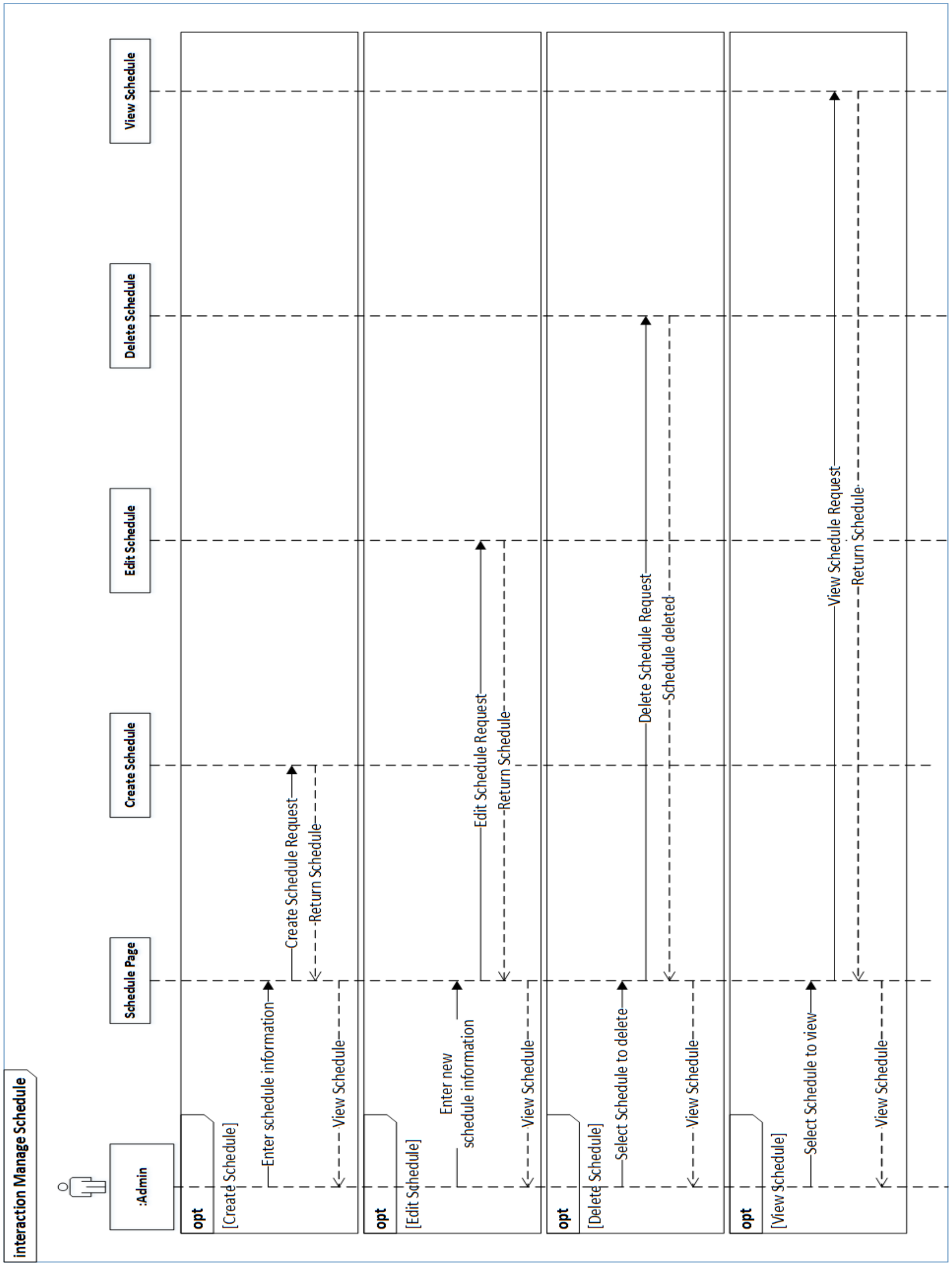


Figure 7: Manage Schedule Sequence Diagram

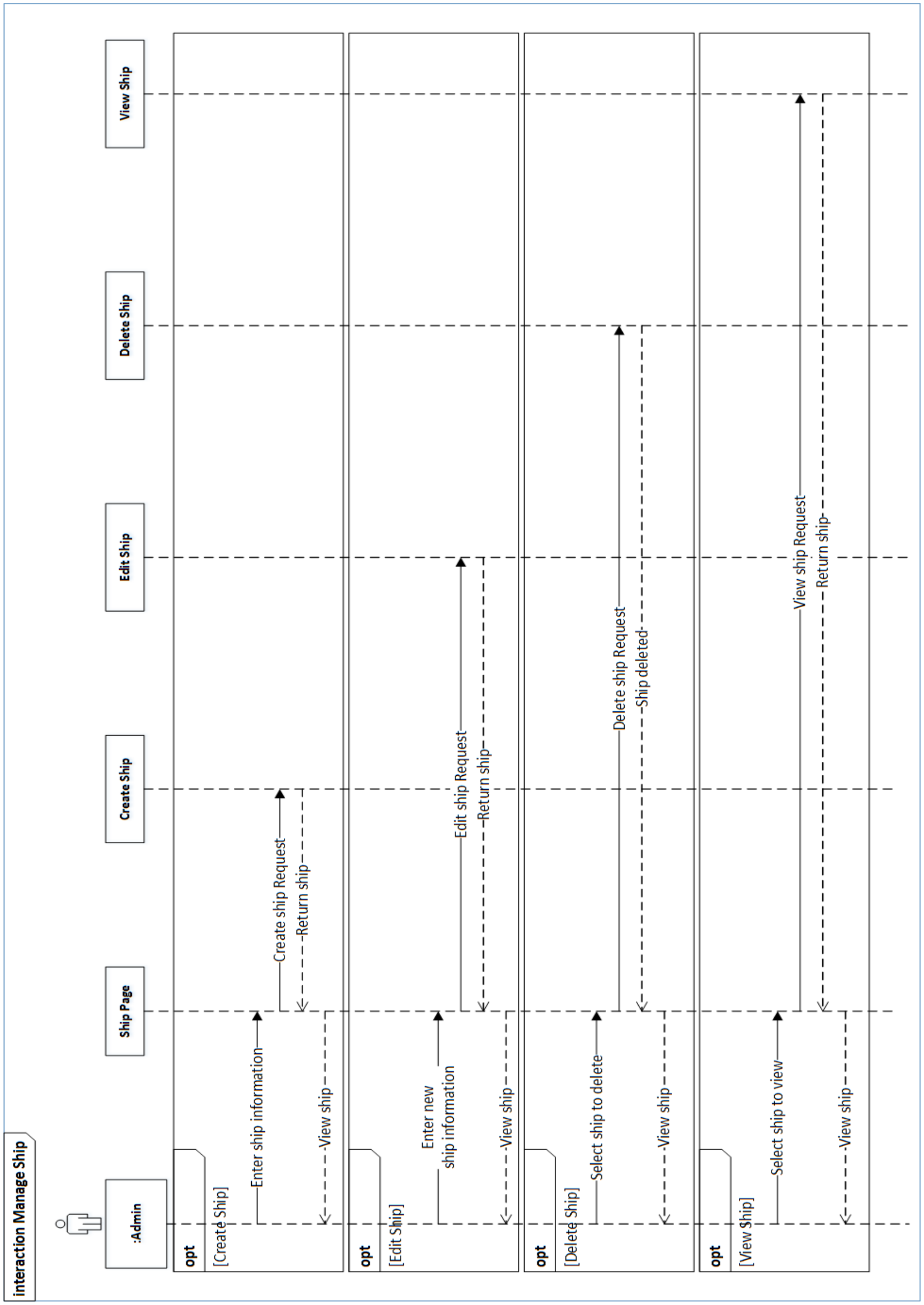


Figure 8: Manage Ship Sequence Diagram

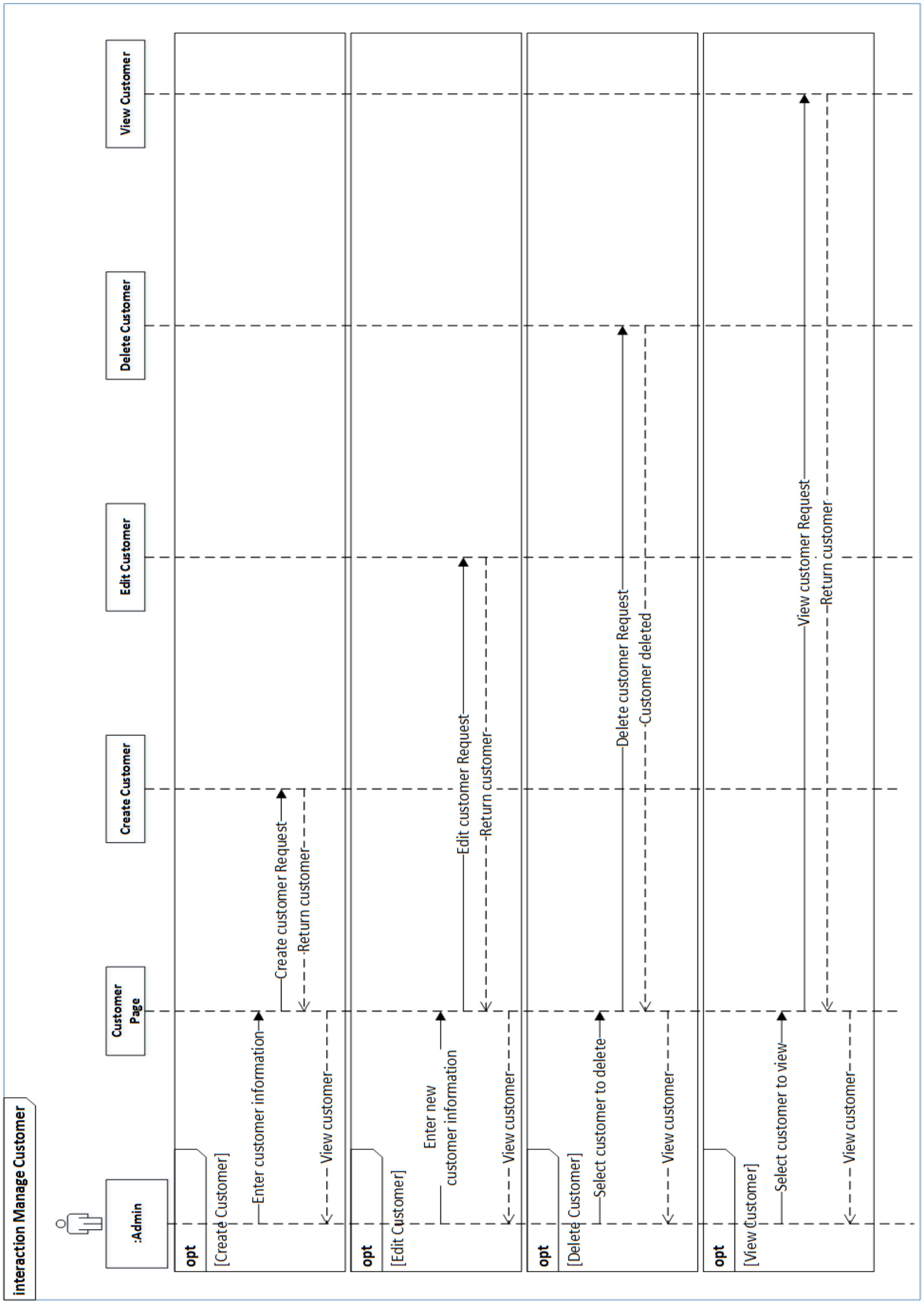


Figure 9: Manage Customer Sequence Diagram

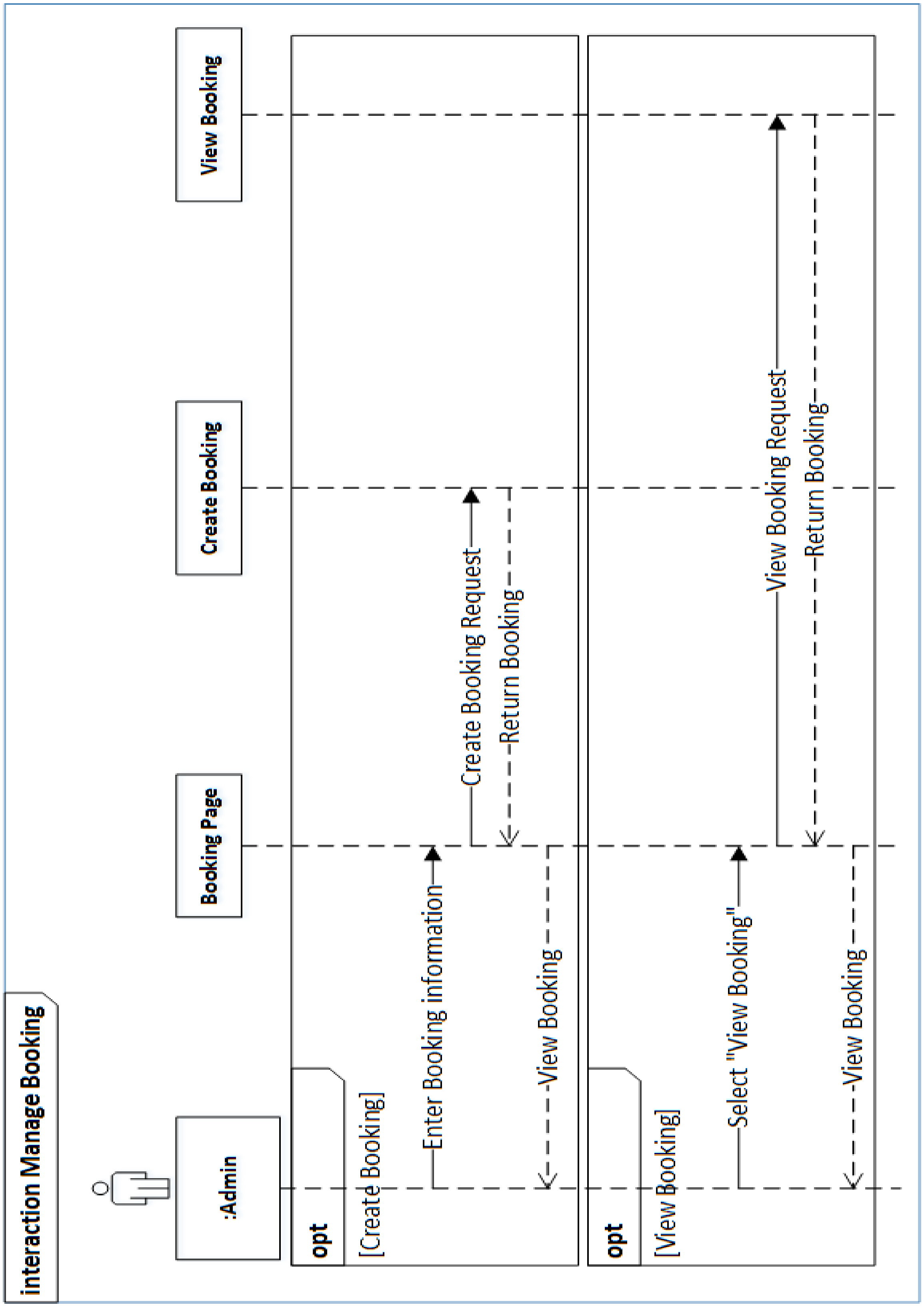


Figure 10: Manage Booking Sequence Diagram

3.3.4 Class Diagram

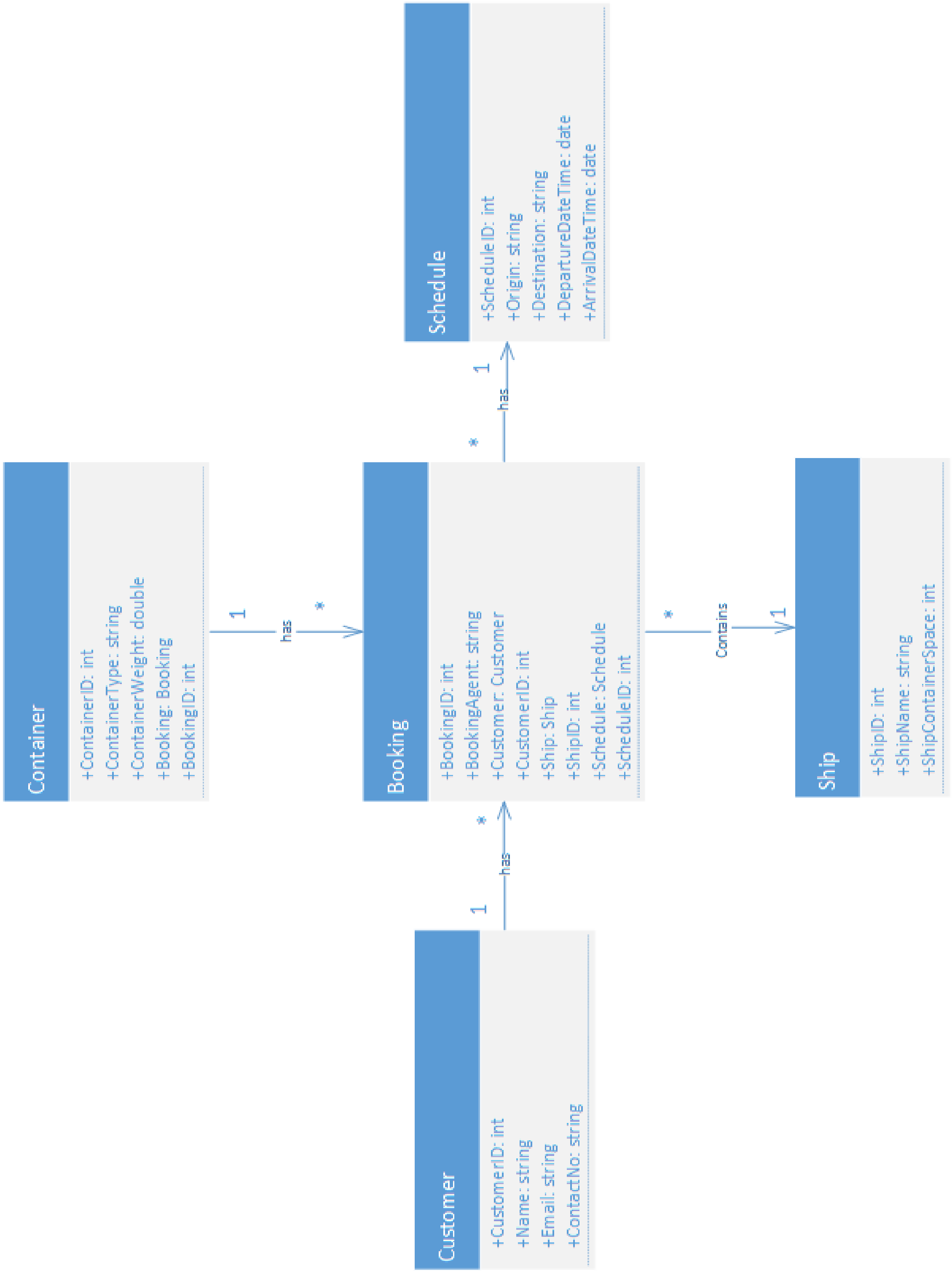
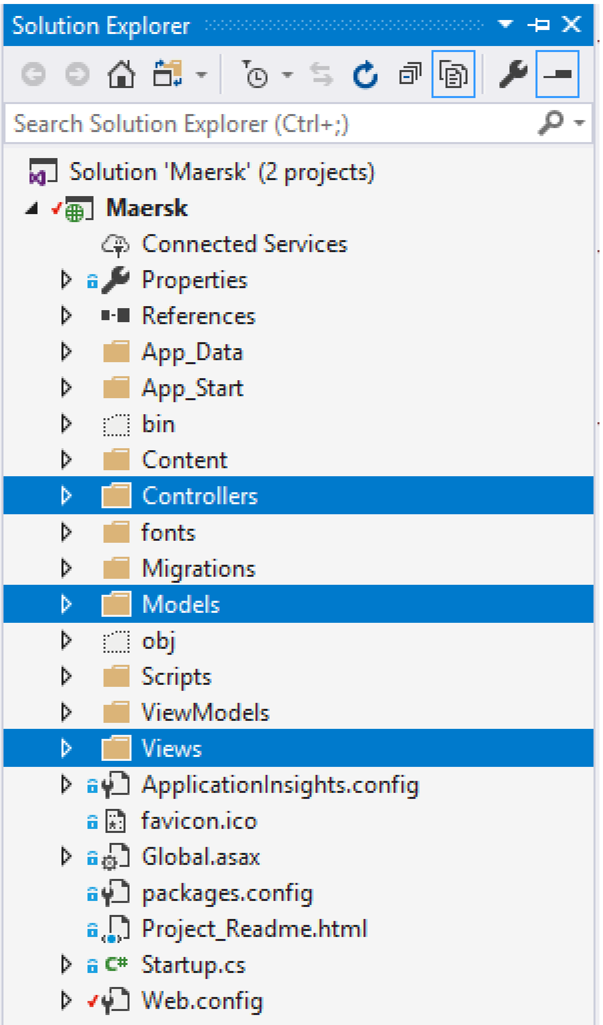


Figure 11: Class Diagram

4.0 Implementation

4.1 ASP.NET Web Application

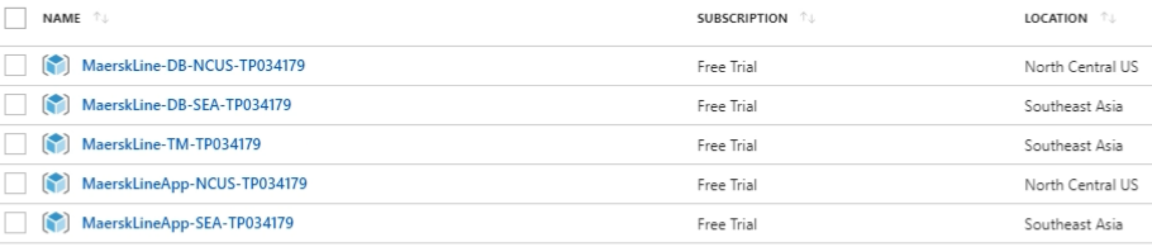


```
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Maersk' (2 projects)
  Maersk
      Connected Services
      Properties
      References
      App_Data
      App_Start
      bin
      Content
      Controllers
      fonts
      Migrations
      Models
      obj
      Scripts
      ViewModels
      Views
      ApplicationInsights.config
      favicon.ico
      Global.asax
      packages.config
      Project_Readme.html
      Startup.cs
      Web.config
```

Figure 12: Maersk Line Project

This Maersk Line project is developed in web based by using ASP.NET with the MVC5 framework; MVC is model, view, controller. The purpose of using MVC framework to develop websites is because it is easy to maintain, test, and develop as it has separate the codes in 3 difference components which are MVC. Moreover, the Maersk Line project is using Bootstrap version 3.3.7 where it is a front-end framework which allows users to view the website in a very good appearance. It can also auto scale to different screen size of the user's devices and allow users to view website properly.

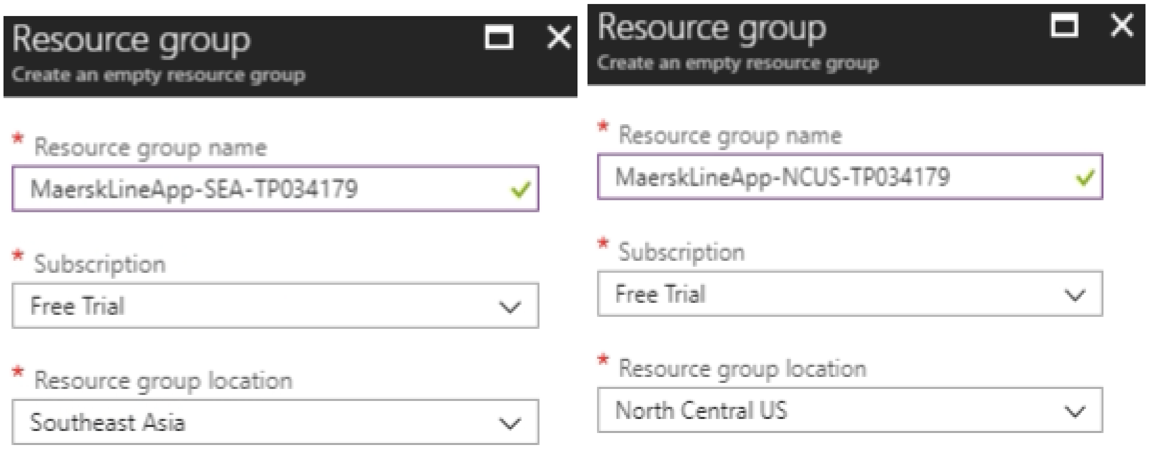
4.2 Azure Resource Group



NAME	SUBSCRIPTION	LOCATION
MaerskLine-DB-NCUS-TP034179	Free Trial	North Central US
MaerskLine-DB-SEA-TP034179	Free Trial	Southeast Asia
MaerskLine-TM-TP034179	Free Trial	Southeast Asia
MaerskLineApp-NCUS-TP034179	Free Trial	North Central US
MaerskLineApp-SEA-TP034179	Free Trial	Southeast Asia

Figure 13: Maersk Line Resource Group

4.2.1 Resource Group of Web Application Service



Resource group
Create an empty resource group

* Resource group name
MaerskLineApp-SEA-TP034179 ✓

* Subscription
Free Trial

* Resource group location
Southeast Asia

Resource group
Create an empty resource group

* Resource group name
MaerskLineApp-NCUS-TP034179 ✓

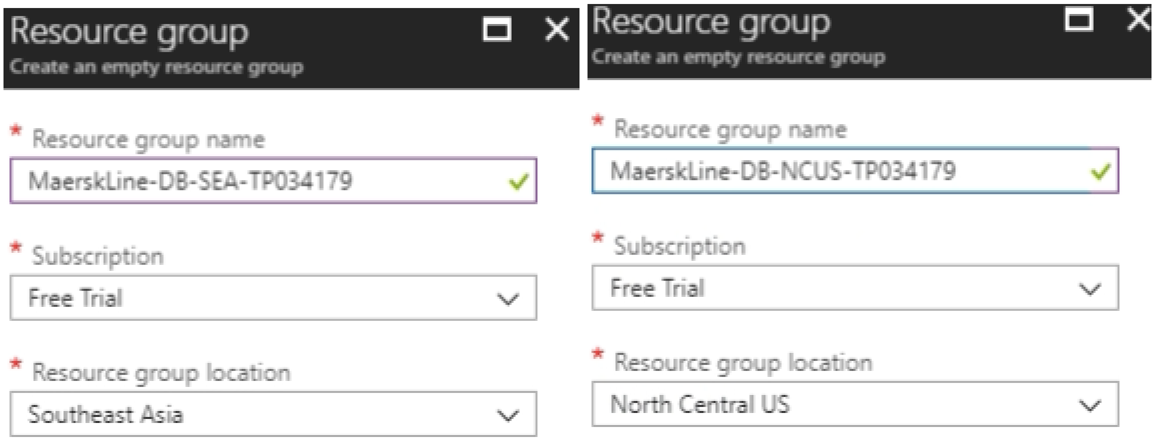
* Subscription
Free Trial

* Resource group location
North Central US

Figure 14: Web App Resource Groups

In Figure 14, there are two resource groups created by developer which are primary and secondary web application. Resource group in Azure is to provide monitoring, control, access, provision, and manage billing for collections of assets that are required to run an application (Russell Smith, 2015). The resource group for Web App is created in two different names as MaerskLineApp-SEA-TP034179 and MaerskLineApp-NCUS-TP034179. The "SEA" is Southeast Asia where "NCUS" is North Central US. In this Maersk Line project, SEA is selected as the primary resource group because most of the primary users for this Maersk Line application are in this region.

4.2.2 Resource Groups of SQL Database



Resource group
Create an empty resource group

* Resource group name
MaerskLine-DB-SEA-TP034179 ✓

* Subscription
Free Trial

* Resource group location
Southeast Asia

Resource group
Create an empty resource group

* Resource group name
MaerskLine-DB-NCUS-TP034179 ✓

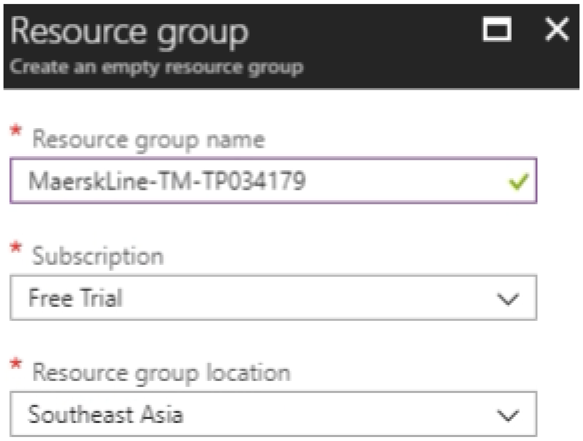
* Subscription
Free Trial

* Resource group location
North Central US

Figure 15: Resource Group for SQL Database

In Figure 15, there are two resource group for SQL Database. They are created in two different names as the purpose of separating the two different regions “SEA” and “NCUS”. The primary database is “SEA”, where secondary database is “NCUS”.

4.2.3 Resource Group of Traffic Manager



Resource group
Create an empty resource group

* Resource group name
MaerskLine-TM-TP034179 ✓

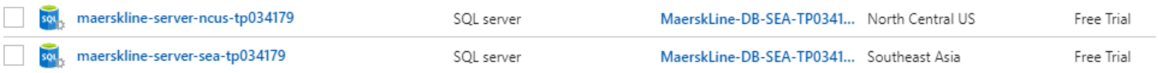
* Subscription
Free Trial

* Resource group location
Southeast Asia

Figure 16: Resource Group for Traffic Manager

In Figure 16, there is a resource group created for traffic manager. Traffic manager will deliver high availability for your applications by monitoring your endpoints and providing automatic failover when an endpoint goes down (Dwivedi, Madureira and Tuliani, 2017).

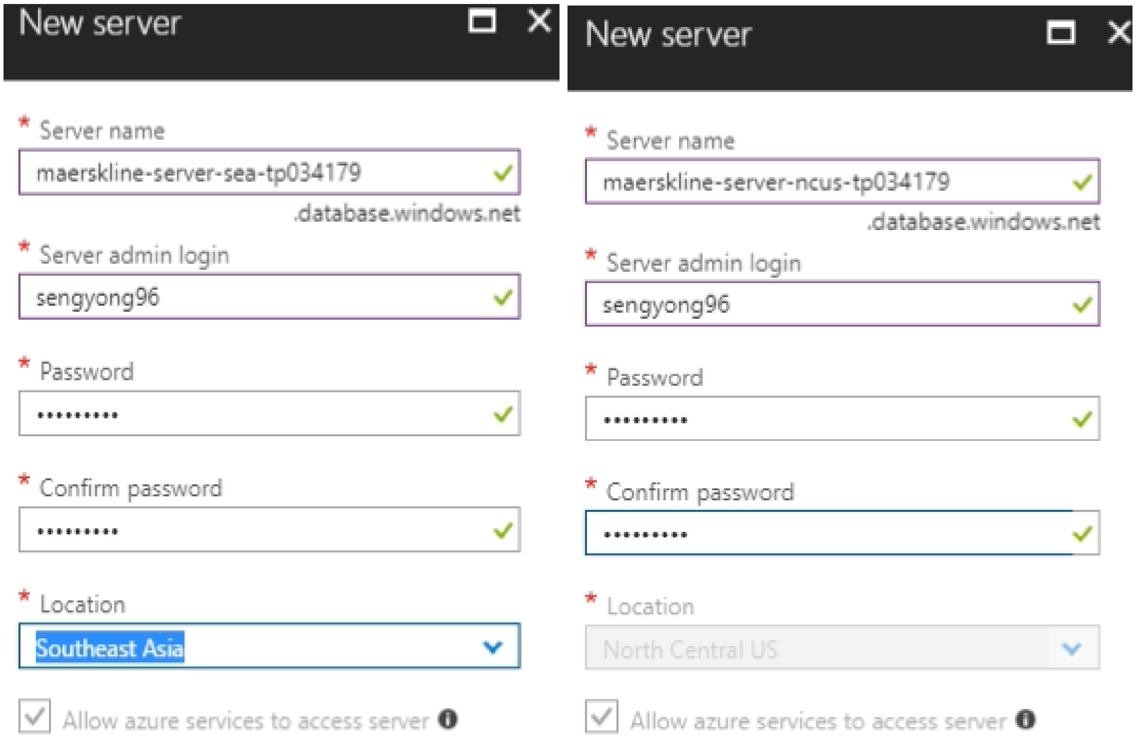
4.3 Azure SQL Server



□ maerskline-server-ncus-tp034179	SQL server	MaerskLine-DB-SEA-TP0341...	North Central US	Free Trial
□ maerskline-server-sea-tp034179	SQL server	MaerskLine-DB-SEA-TP0341...	Southeast Asia	Free Trial

Figure 17: Azure SQL Server

4.3.1 Setup SQL Server



New server

* Server name
maerskline-server-sea-tp034179 ✓
.database.windows.net

* Server admin login
sengyong96 ✓

* Password
••••••••• ✓

* Confirm password
••••••••• ✓

* Location
Southeast Asia

☑ Allow azure services to access server

New server

* Server name
maerskline-server-ncus-tp034179 ✓
.database.windows.net

* Server admin login
sengyong96 ✓

* Password
••••••••• ✓

* Confirm password
••••••••• ✓

* Location
North Central US

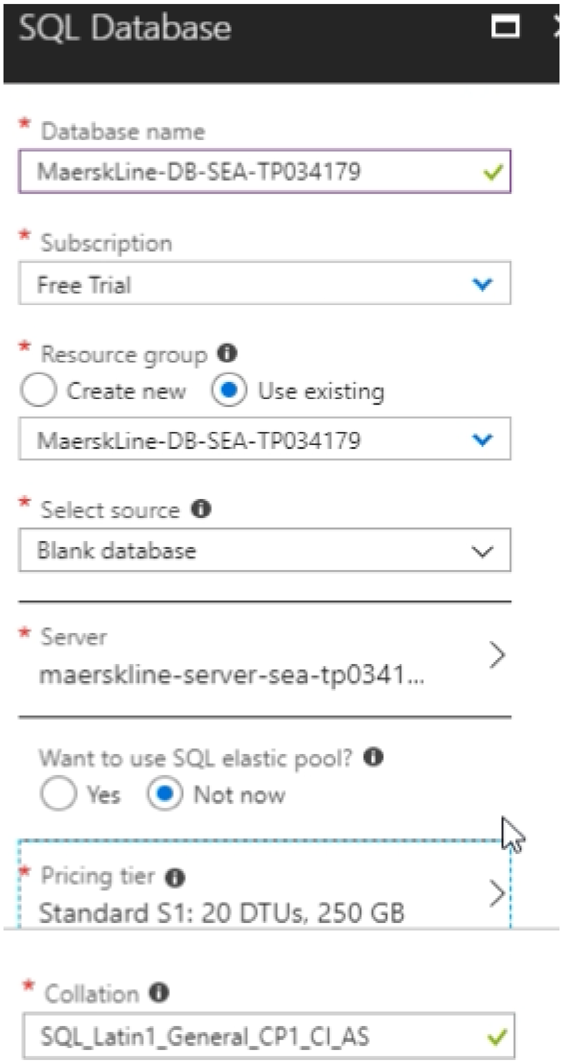
☑ Allow azure services to access server

Figure 18: Setup SQL Server

It is required to setup SQL server before successfully create SQL databases in Azure. The SQL server for Maersk Line TP034179 has created in 2 different names also as it is used to differentiate two different regions, SEA and NCUS.

4.4 Azure SQL Database

4.4.1 Setup Primary SQL Database



SQL Database

* Database name
MaerskLine-DB-SEA-TP034179 ✓

* Subscription
Free Trial

* Resource group
○ Create new ● Use existing
MaerskLine-DB-SEA-TP034179

* Select source
Blank database

* Server
maerskline-server-sea-tp0341...

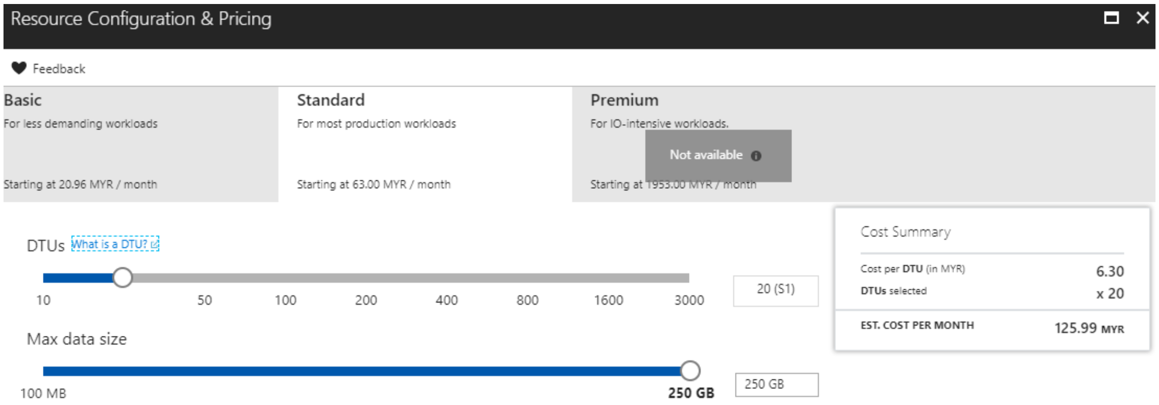
Want to use SQL elastic pool?
○ Yes ● Not now

* Pricing tier
Standard S1: 20 DTUs, 250 GB

* Collation
SQL_Latin1_General_CP1_CI_AS ✓

Figure 19: Setup Primary SQL Database

By creating Primary SQL database successfully, the SQL server must be setup and linked to this SQL database. The primary database is created as MaerskLine-DB-SEA-TP034179, the source of the primary database is “Blank database”.


Resource Configuration & Pricing

Feedback

Basic
For less demanding workloads
Starting at 20.96 MYR / month

Standard
For most production workloads
Starting at 63.00 MYR / month

Premium
For IO-intensive workloads.
Not available
Starting at 1953.00 MYR / month

DTUs What is a DTU?
20 (S1)

Max data size
250 GB

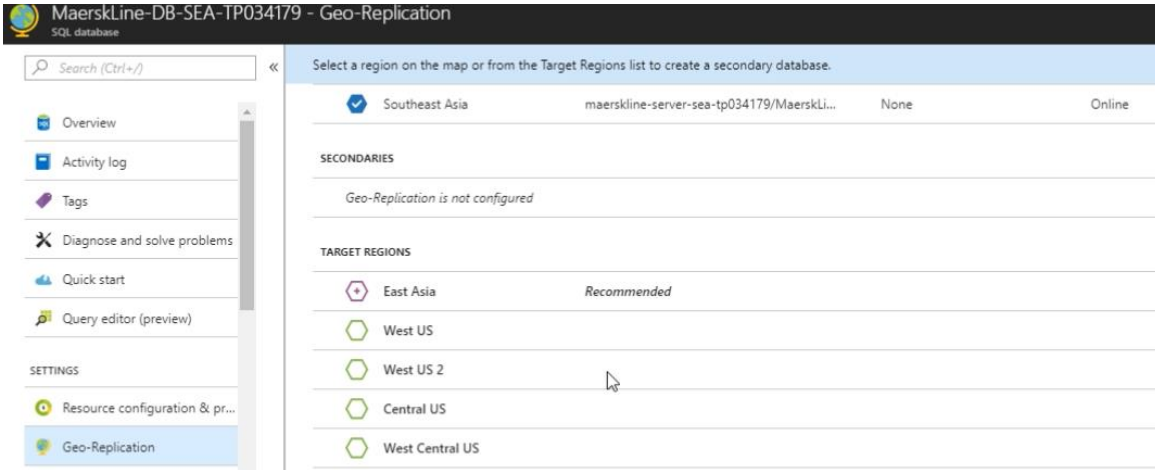
Cost Summary	
Cost per DTU (in MYR)	6.30
DTUs selected	x 20
EST. COST PER MONTH	125.99 MYR

Figure 20: Standard (S1) pricing tier

Standard (S1) pricing tier is selected for the primary SQL database for SEA region. It has included 20 Database Transaction Units (DTUs) and maximum data size of 250 GB. The system can perform 20 users concurrently in every one second where the DTUs has 20 units.

4.4.2 Setup Secondary SQL Database

4.4.2.1 Replicate Secondary Database


MaerskLine-DB-SEA-TP034179 - Geo-Replication
SQL database

Select a region on the map or from the Target Regions list to create a secondary database.

Southeast Asia	maerskline-server-sea-tp034179/MaerskLi...	None	Online

SECONDARIES

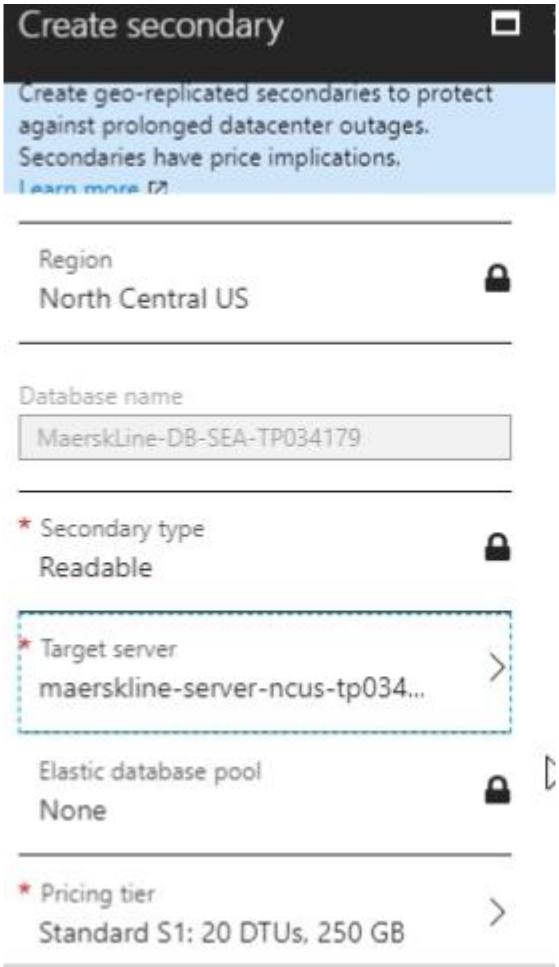
Geo-Replication is not configured

TARGET REGIONS

East Asia	Recommended
West US	
West US 2	
Central US	
West Central US	

Figure 21: Replicate Secondary Database

There is a service called “Geo-Replication” in Azure SQL database for creating a copy of the database to a different region. In this case, SQL database in SEA region will be replicate as secondary SQL database in NCUS region. The purpose of replicating the database is to use for failover. The meaning of failover is when the primary database is unable to access or broken, the secondary database will be replaced and it will become the primary database for that moment.



Create secondary

Create geo-replicated secondaries to protect against prolonged datacenter outages. Secondaries have price implications.

Learn more

Region: North Central US

Database name: MaerskLine-DB-SEA-TP034179

* Secondary type: Readable

* Target server: maerskline-server-ncus-tp034...

Elastic database pool: None

* Pricing tier: Standard S1: 20 DTUs, 250 GB

Figure 22: Setup Secondary SQL Database

The secondary SQL database for Maersk Line TP034179 is created by selecting the “Target Regions” which is shown in Figure 22. The secondary database is created at NCUS region. The secondary type for the database is “readable” and the Target Server is the server created earlier. The pricing tier is the same by default with the primary SQL database which has created earlier too.

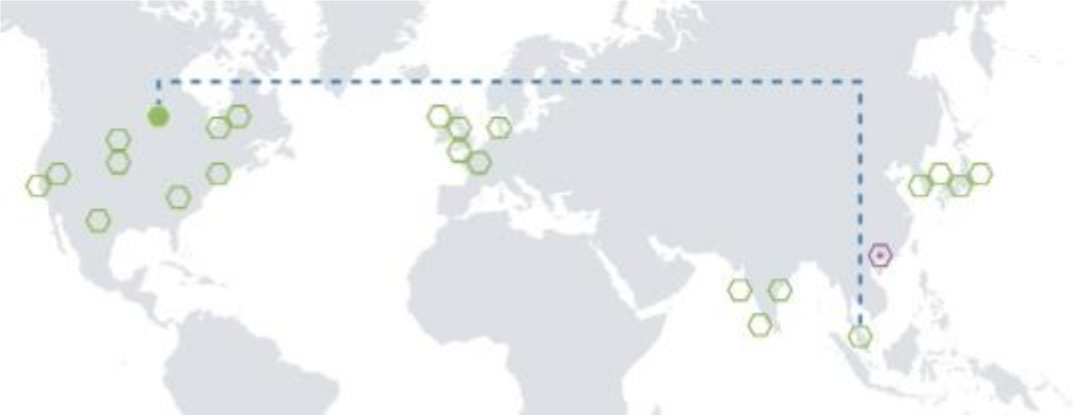


Figure 23: Seeding Secondary Database

After creating the secondary database, it will be seeding the secondary database with the primary database in the world map as shown in this figure above.

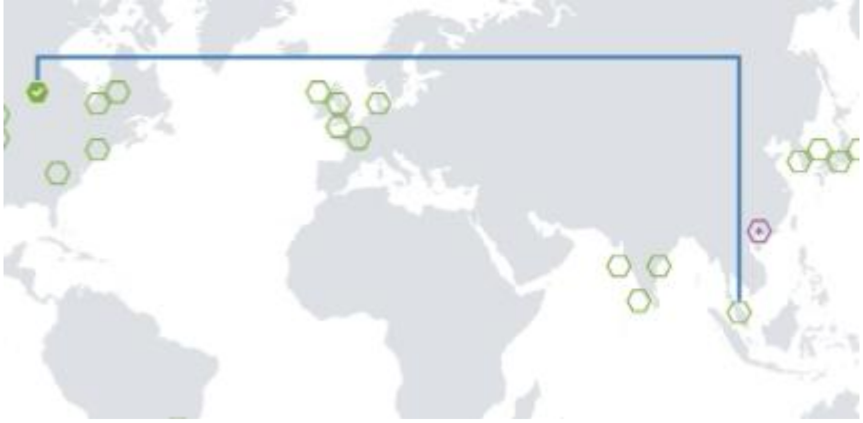
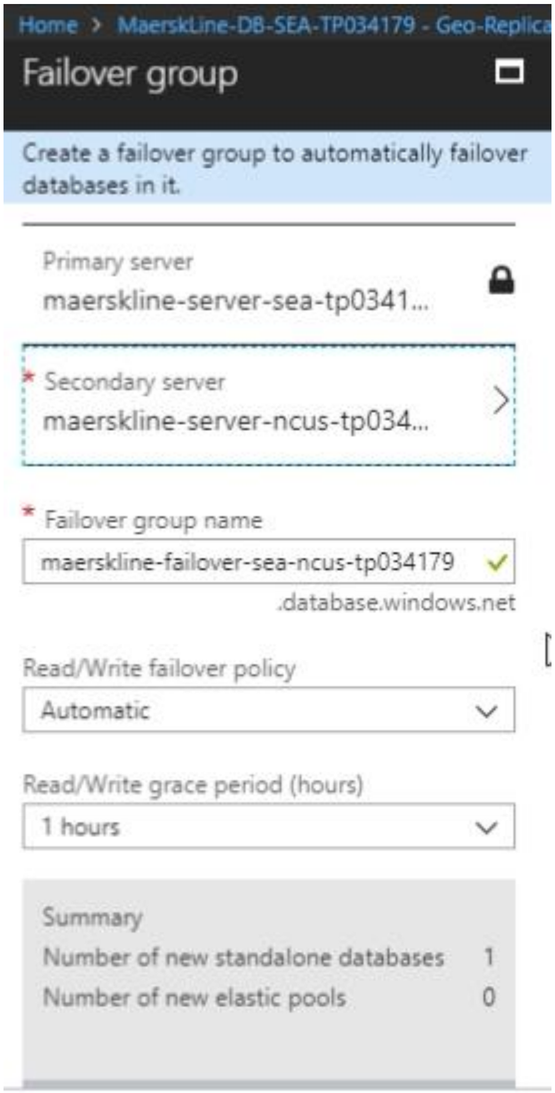


Figure 24: Successfully seeding secondary database

The lines of seeding secondary database with the primary database is changed into solid line after the replication process is completed successfully.

4.4.2.2 Configure Failover Policy



Home > MaerskLine-DB-SEA-TP034179 - Geo-Replica

Failover group

Create a failover group to automatically failover databases in it.

Primary server
maerskline-server-sea-tp0341...

* Secondary server
maerskline-server-ncus-tp034...

* Failover group name
maerskline-failover-sea-ncus-tp034179 ✓
.database.windows.net

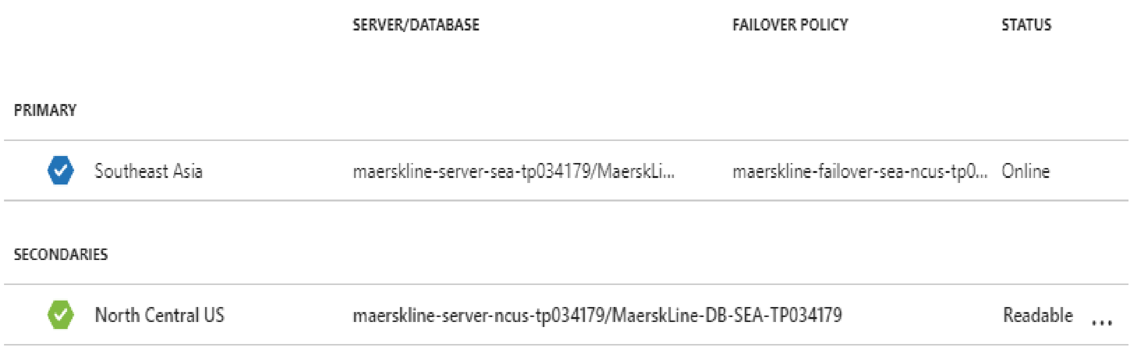
Read/Write failover policy
Automatic

Read/Write grace period (hours)
1 hours

Summary
Number of new standalone databases 1
Number of new elastic pools 0

Figure 25: Configure Failover Policy

After replicate the secondary SQL database from the primary SQL database, there will be a configuration process for failover policy. By doing so, click on the “Failover groups” at the Setting, and select “Add Group” in the primary SQL server for primary SQL database. The failover group is named as maerskline-failover-sea-ncus-tp034179.database.windows.net. The other settings are set as default which is automatic for read/write failover policy and read/write grace period (hours) is 1 hour.

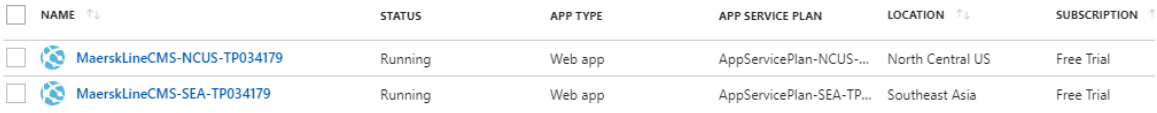


	SERVER/DATABASE	FAILOVER POLICY	STATUS
PRIMARY			
✓ Southeast Asia	maerskline-server-sea-tp034179/MaerskLi...	maerskline-failover-sea-ncus-tp0...	Online
SECONDARIES			
✓ North Central US	maerskline-server-ncus-tp034179/MaerskLine-DB-SEA-TP034179		Readable ...

Figure 26: SQL database Geo-Replication

After the failover group is successfully setup, the primary SQL database will have a column name as failover policy and has the failover name as created earlier will be showing at here, "maerskline-failover-sea-ncus-tp0…".

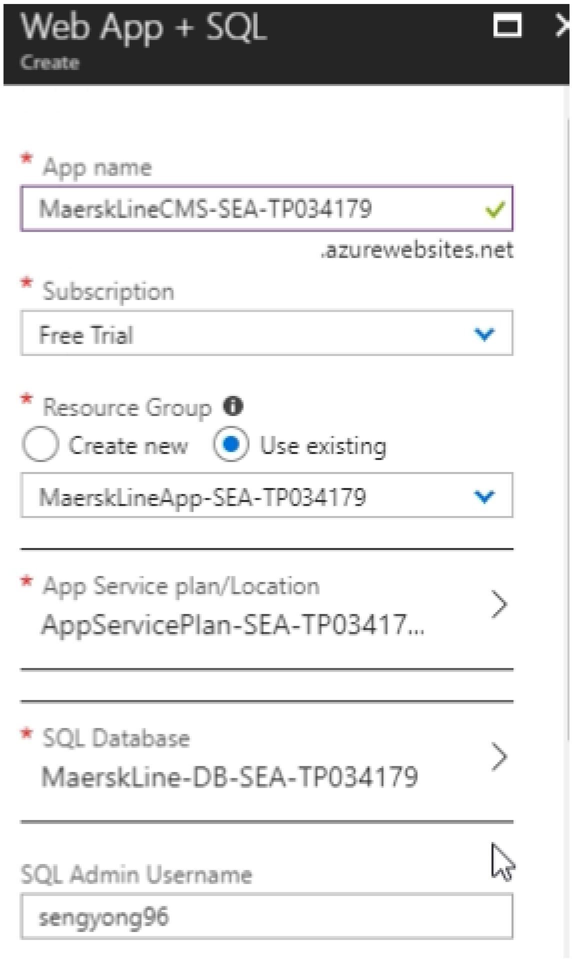
4.5 Azure Web Application Service



□ NAME ↑↓	STATUS	APP TYPE	APP SERVICE PLAN	LOCATION ↑↓	SUBSCRIPTION ↑
□ MaerskLineCMS-NCUS-TP034179	Running	Web app	AppServicePlan-NCUS-...	North Central US	Free Trial
□ MaerskLineCMS-SEA-TP034179	Running	Web app	AppServicePlan-SEA-TP...	Southeast Asia	Free Trial

Figure 27: Azure Web Application Service

Azure Web Application Service is the service for hosting web applications, REST APIs, and mobile back ends (Lin et al., 2017).



Web App + SQL
Create

* App name
MaerskLineCMS-SEA-TP034179 ✓
.azurewebsites.net

* Subscription
Free Trial

* Resource Group
○ Create new ● Use existing
MaerskLineApp-SEA-TP034179

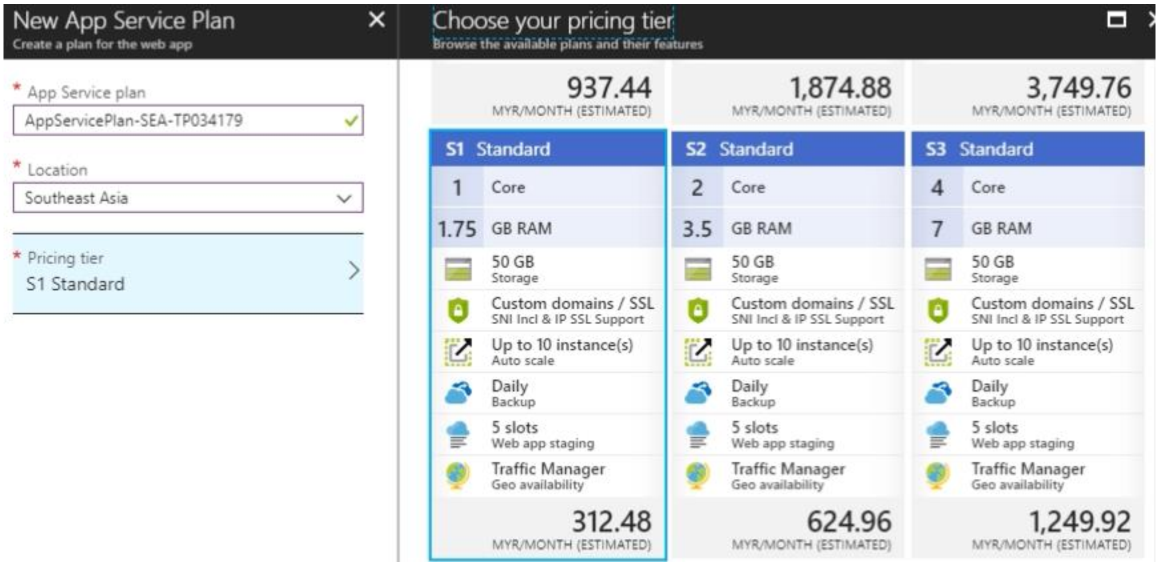
* App Service plan/Location
AppServicePlan-SEA-TP03417...

* SQL Database
MaerskLine-DB-SEA-TP034179

SQL Admin Username
sengyong96

Figure 28: Web App Service for Maersk Line

The Web App for Maersk Line in SEA region is named as shown in "MaerskLineCMS-SEA-TP034179.azurewebsites.net" which is shown in the figure above. This will be the URL of the Maersk Line project for SEA region. By creating the Maersk Line project for NCUS region, it will be the same procedures for creating the SEA region. The URL of NCUS region for the Web App will be "MaerskLineCMS-NCUS-TP034179.azurewebsites.net". The resource group will be using the resource that has created earlier with the same region as the Web App region.



New App Service Plan
Create a plan for the web app

* App Service plan
AppServicePlan-SEA-TP034179

* Location
Southeast Asia

* Pricing tier
S1 Standard

Choose your pricing tier
Browse the available plans and their features

937.44 MYR/MONTH (ESTIMATED)	1,874.88 MYR/MONTH (ESTIMATED)	3,749.76 MYR/MONTH (ESTIMATED)
S1 Standard	S2 Standard	S3 Standard
1 Core	2 Core	4 Core
1.75 GB RAM	3.5 GB RAM	7 GB RAM
50 GB Storage	50 GB Storage	50 GB Storage
Custom domains / SSL SNI Incl & IP SSL Support	Custom domains / SSL SNI Incl & IP SSL Support	Custom domains / SSL SNI Incl & IP SSL Support
Up to 10 instance(s) Auto scale	Up to 10 instance(s) Auto scale	Up to 10 instance(s) Auto scale
Daily Backup	Daily Backup	Daily Backup
5 slots Web app staging	5 slots Web app staging	5 slots Web app staging
Traffic Manager Geo availability	Traffic Manager Geo availability	Traffic Manager Geo availability
312.48 MYR/MONTH (ESTIMATED)	624.96 MYR/MONTH (ESTIMATED)	1,249.92 MYR/MONTH (ESTIMATED)

Figure 29: Web App Service Plan

The App Service Plan for SEA region will be selecting S1 Standard pricing tier. The specifications for the App Service Plan S1 is sufficient for SEA region as there has 1 Core, 1.75 GB RAM, 50 GB storage, 5 slots Web app staging, Traffic Manager and etc with the estimated price of MYR 312.48 per month. As for the NCUS region, by repeating the steps of creating SEA region's Web App and App Service Plan will do as mentioned earlier.

4.5.2 Deploy ASP.NET Web Application

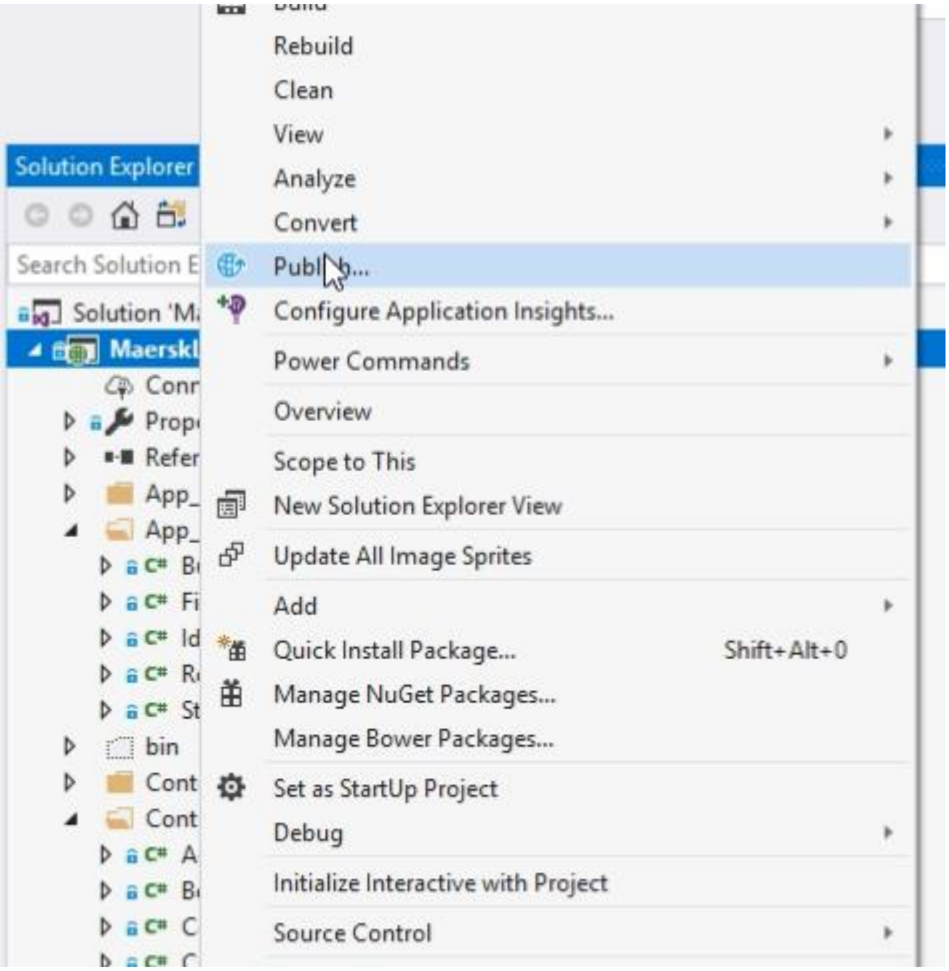
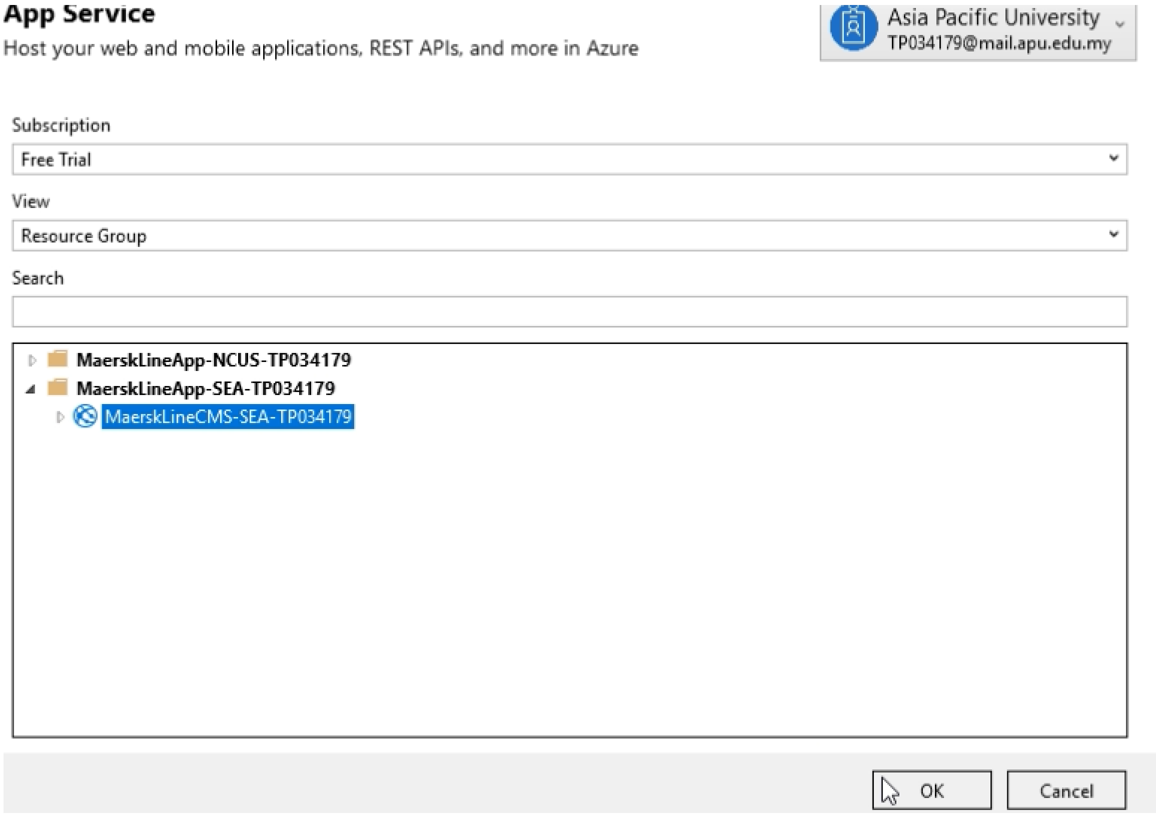


Figure 30: Publishing MaerskLine-TP034179

By publishing the Maersk Line project, the previous process must be done before publishing the Maersk Line project into Azure cloud platform. By right clicking the project name, select "Publish" as shown in Figure above, the project will be publishing.


App Service
Host your web and mobile applications, REST APIs, and more in Azure
Asia Pacific University
TP034179@mail.apu.edu.my
OK
Cancel

Subscription: Free Trial

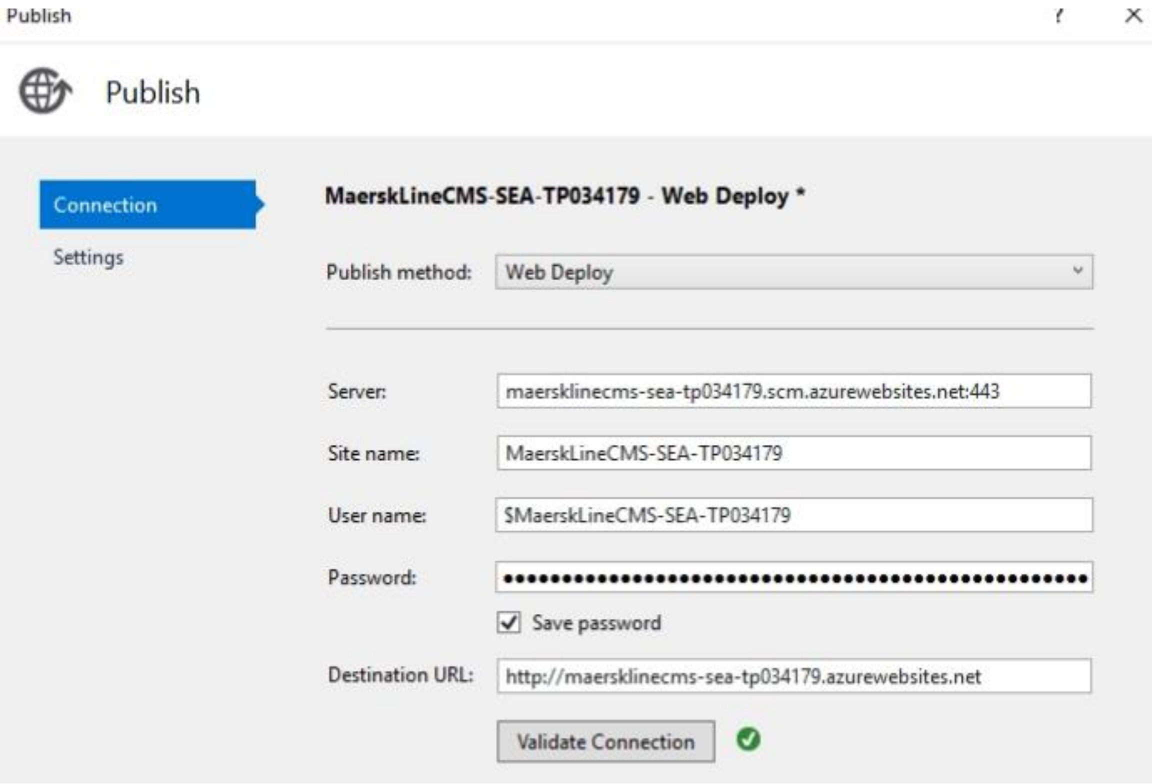
View: Resource Group

Search:

- MaerskLineApp-NCUS-TP034179
- MaerskLineApp-SEA-TP034179
  - MaerskLineCMS-SEA-TP034179

Figure 31: Select Azure Web App

By publishing the Maersk Line project for SEA region, select the Azure Web App for SEA region, and press “OK”. The SEA region web app is “MaerskLineCMS-SEA-TP034179”.



Publish

Publish

MaerskLineCMS-SEA-TP034179 - Web Deploy *

Publish method: Web Deploy

Server: maersklinecms-sea-tp034179.scm.azurewebsites.net:443

Site name: MaerskLineCMS-SEA-TP034179

User name: $MaerskLineCMS-SEA-TP034179

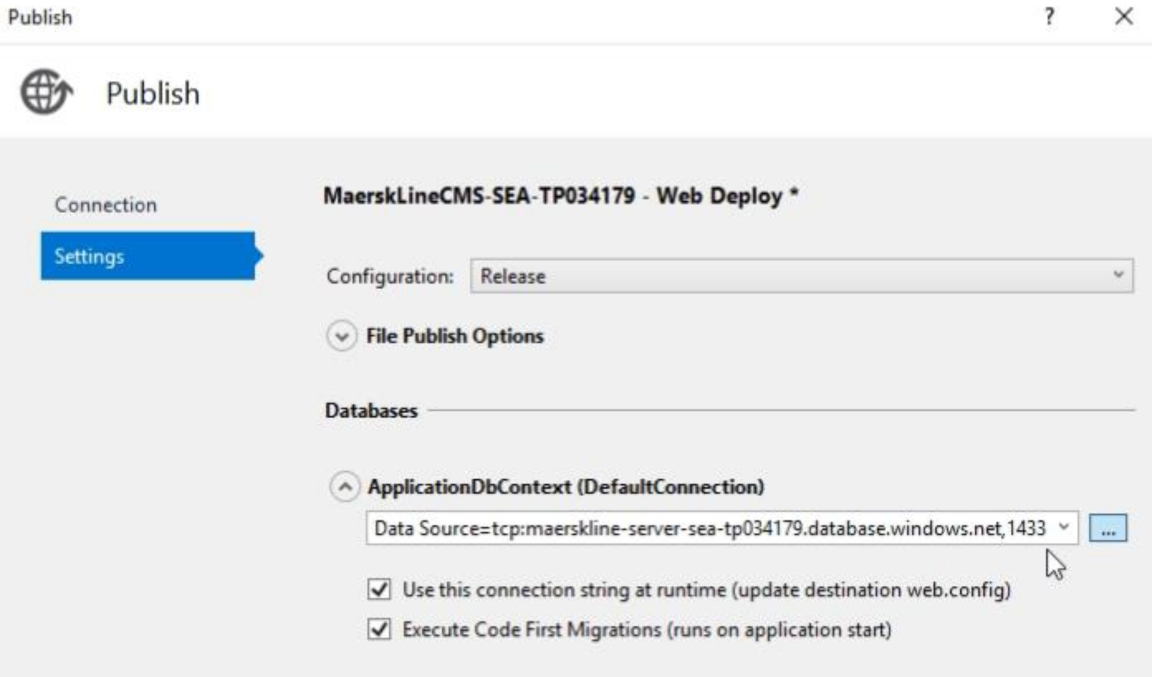
Password: ●●

☑ Save password

Destination URL: http://maersklinecms-sea-tp034179.azurewebsites.net

Figure 32: Azure Web App Connection

After pressing “OK”, the system will redirect user to this page for setting up the connection and pressing “Validate Connection” to ensure the connection is valid for publish.



MaerskLineCMS-SEA-TP034179 - Web Deploy *

Configuration: Release

File Publish Options

Databases

ApplicationDbContext (DefaultConnection)

Data Source=tcp:maerskline-server-sea-tp034179.database.windows.net,1433

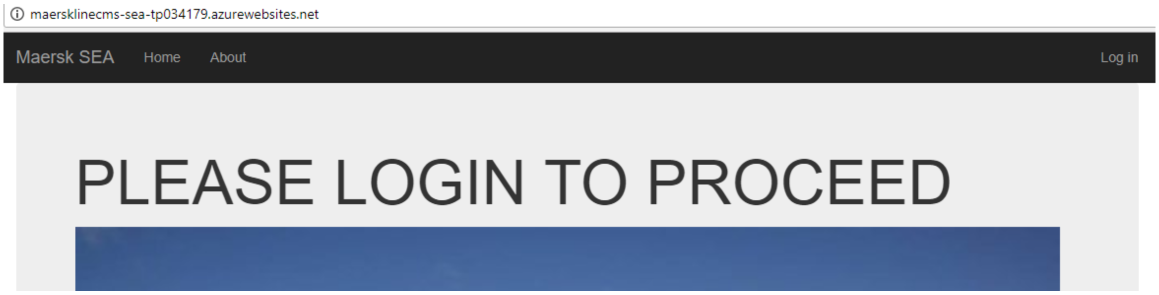
☑ Use this connection string at runtime (update destination web.config)

☑ Execute Code First Migrations (runs on application start)

Figure 33: Azure Web App Settings

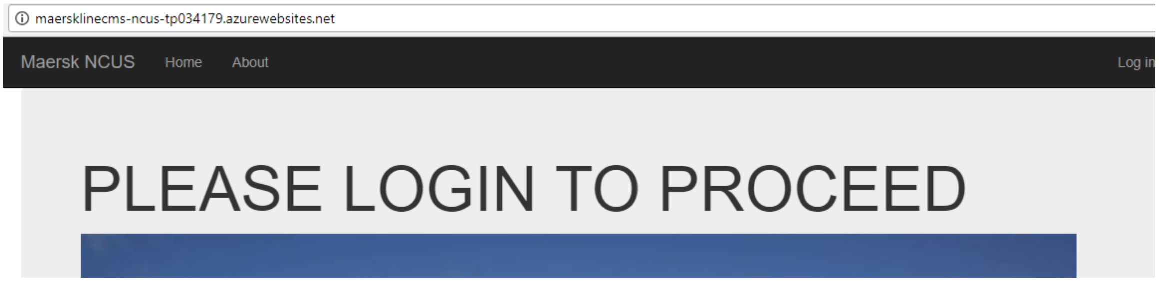
In this page, select the Configuration as “Release”. Then select “Execute Code First Migrations (runs on application start)”. It will allow web application to perform all the migrations once the application is started. The reason of developer is selecting this option is because the Maersk Line project has developed in a code first migration way.

For publishing the NCUS region’s Maersk Line project, it is repeating the steps for SEA region’s Maersk Line project. The final URL for these two regions are “MaerskLineCMS-SEA-TP034179.azurewebsites.net”, and “MaerskLineCMS-NCUS-TP034179.azurewebsites.net”.



PLEASE LOGIN TO PROCEED

Figure 34: Web Application for SEA Region

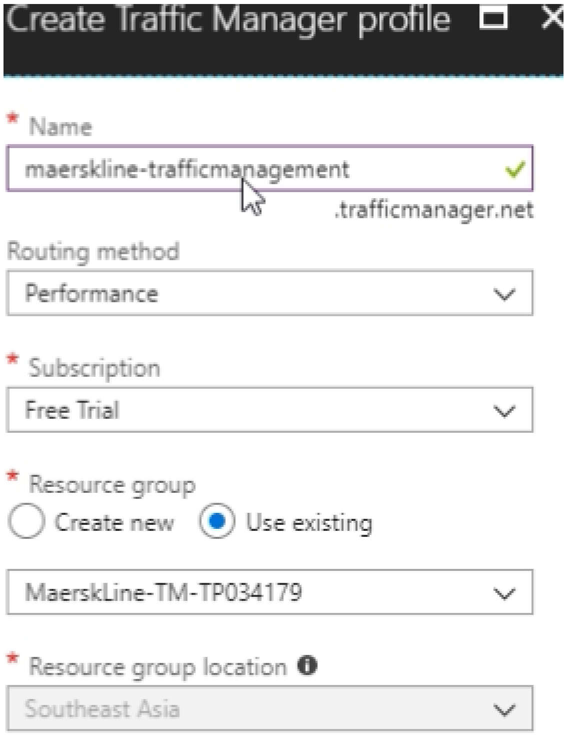


PLEASE LOGIN TO PROCEED

Figure 35: Web Application for NCUS Region

4.6 Azure Traffic Manager

4.6.1 Setup Traffic Manager Profile



Create Traffic Manager profile

* Name

maerskline-trafficmanagement .trafficmanager.net

Routing method

Performance

* Subscription

Free Trial

* Resource group

○ Create new ● Use existing

MaerskLine-TM-TP034179

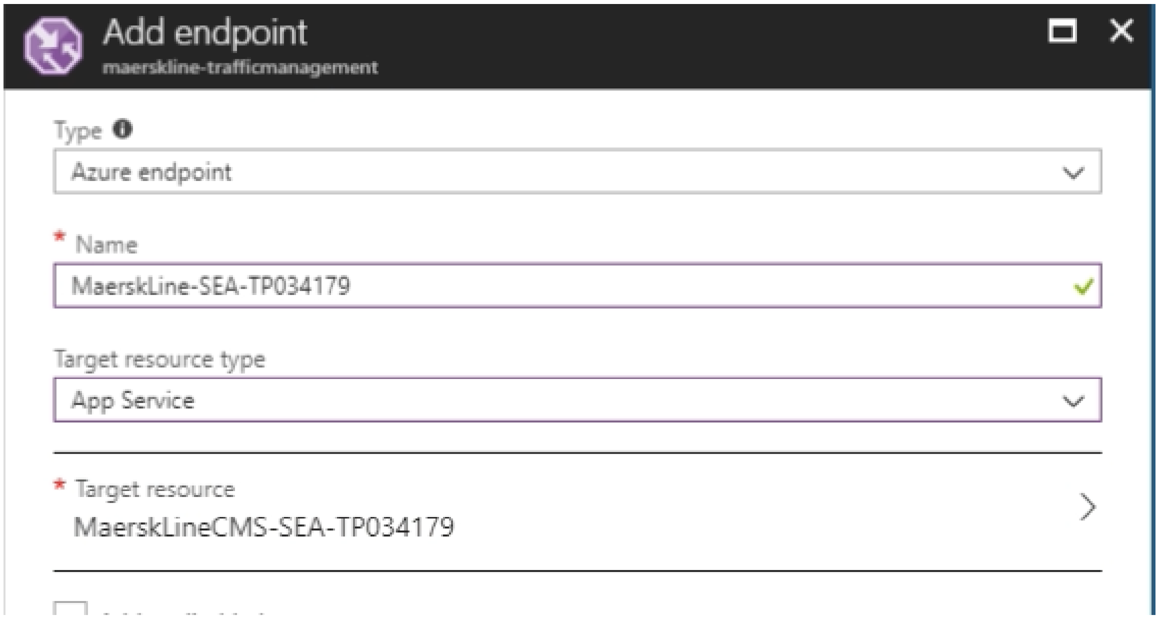
* Resource group location

Southeast Asia

Figure 36: Setup Traffic Manager Profile

By setting up the traffic manager profile, go to search “Traffic Manager Profile” and select “Add” to create traffic manager profile. The naming of this project for the traffic manager profile is “maerskline-trafficmanagement.trafficmanager.net”. The naming of it must be UNIQUE to all other projects. The Routing Method will be selecting “Performance” as it will be managing the traffic based on the system’s performance, the Resource group will be using the existing resource that has created earlier. The purpose of creating it is to increase the responsiveness of the system by routing the traffic of the web application to either SEA or NCUS. The measurement of the system performance is based on the network latency between user and the system.

4.6.2 Setup Endpoints



Add endpoint
maerskline-trafficmanagement

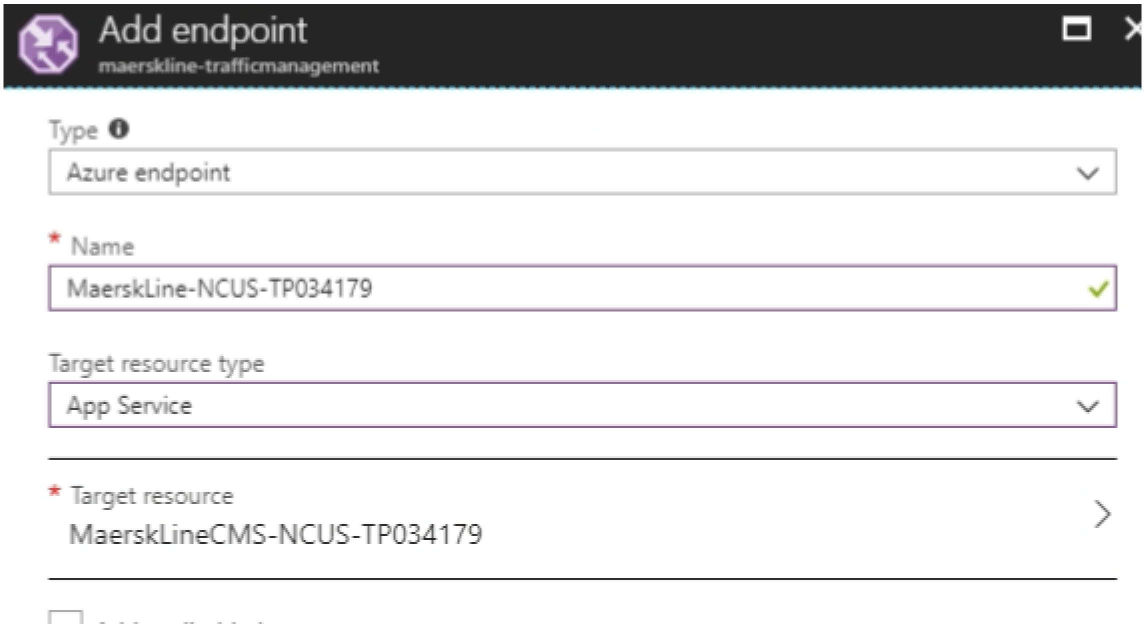
Type: Azure endpoint

* Name: MaerskLine-SEA-TP034179

Target resource type: App Service

* Target resource: MaerskLineCMS-SEA-TP034179

Figure 37: Adding Endpoint for SEA Region



Add endpoint
maerskline-trafficmanagement

Type: Azure endpoint

* Name: MaerskLine-NCUS-TP034179

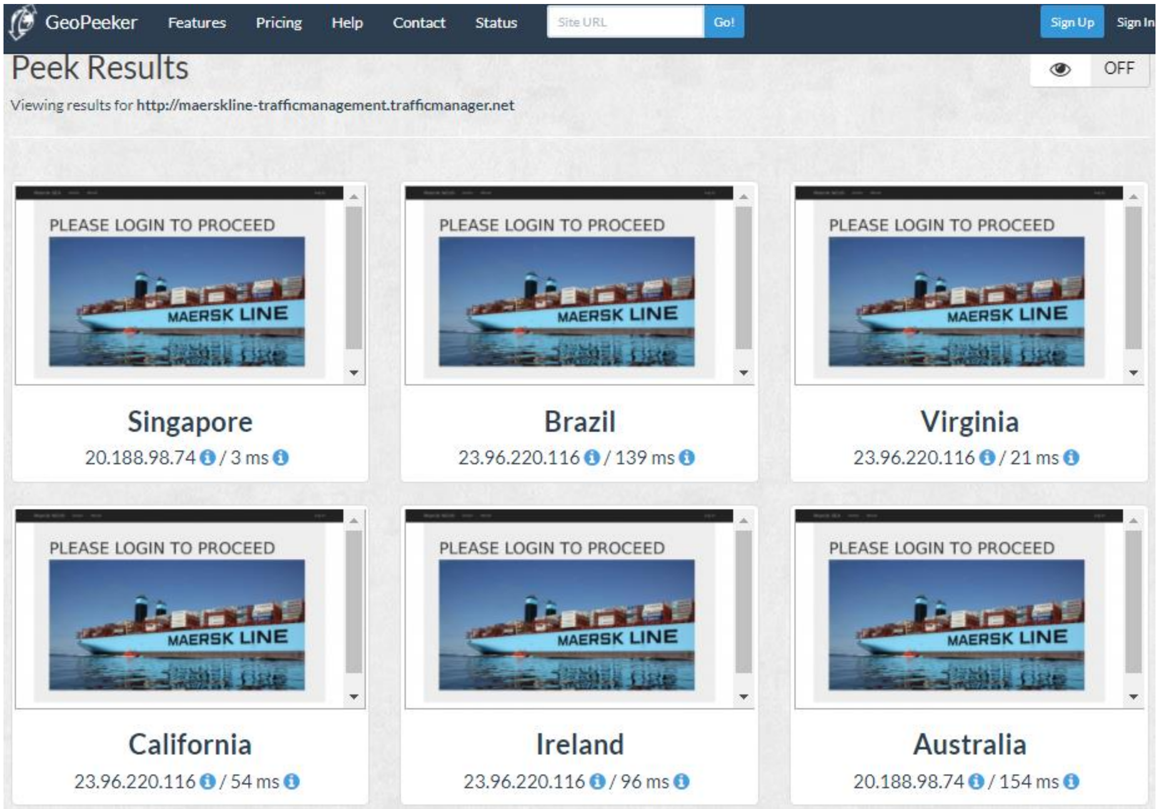
Target resource type: App Service

* Target resource: MaerskLineCMS-NCUS-TP034179

Figure 38: Adding Endpoint for NCUS Region

The endpoints for SEA and NCUS region are created for the traffic manager to distribute the users based on the network latency of the users and the system. By adding the endpoints for both regions, click on the Traffic Manager Profile that has created earlier, go to Settings, and select “Endpoints” to add the endpoints for both regions. The Target resource is selecting the Web App Service of the regions.

4.6.3 Testing Traffic Manager & Endpoints



Peek Results

Viewing results for http://maerskline-trafficmanagement.trafficmanager.net

Singapore
20.188.98.74 / 3 ms

Brazil
23.96.220.116 / 139 ms

Virginia
23.96.220.116 / 21 ms

California
23.96.220.116 / 54 ms

Ireland
23.96.220.116 / 96 ms

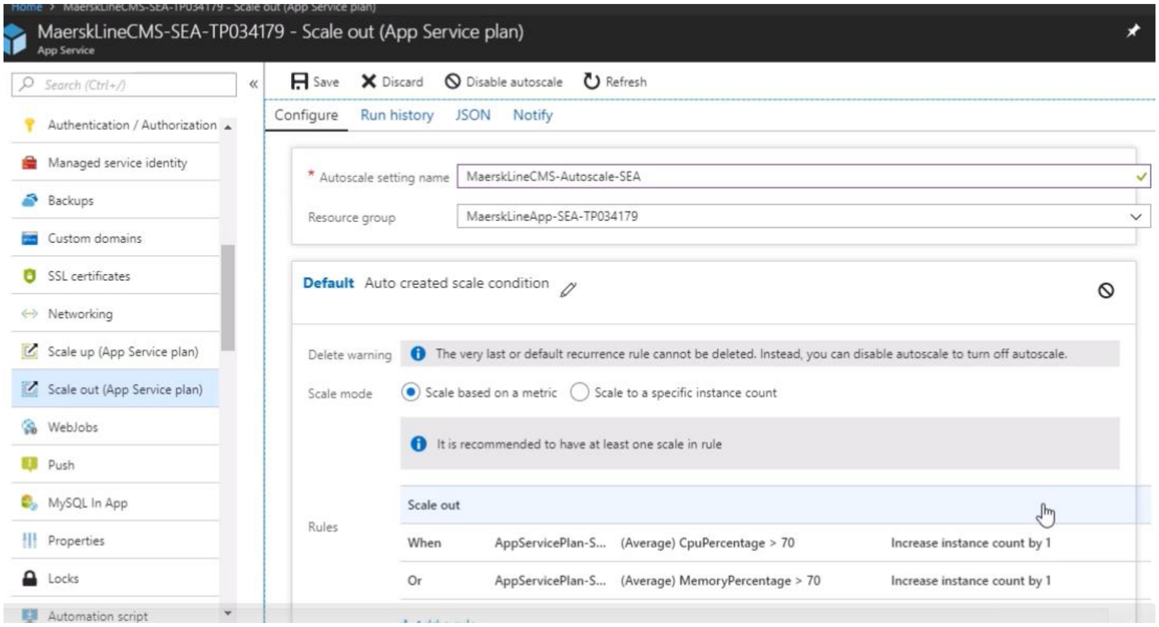
Australia
20.188.98.74 / 154 ms

Figure 39: GeoPeeker for testing the traffic and Endpoints

In the figure above, it has shown that the Internet Protocol (IP) address of Singapore, Australia is the same as because they are fall under SEA regions where Brazil, Virginia, California, Ireland are fall under NCUS regions. Developer has tested these results by using GeoPeeker, https://www.geopeeker.com/.

4.7 Azure Web Application Autoscaling

4.7.1 Setup Web Application Scale Out



MaerskLineCMS-SEA-TP034179 - Scale out (App Service plan)

* Autoscale setting name: MaerskLineCMS-Autoscale-SEA

Resource group: MaerskLineApp-SEA-TP034179

Default Auto created scale condition

Delete warning: The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.

Scale mode: ● Scale based on a metric ○ Scale to a specific instance count

It is recommended to have at least one scale in rule

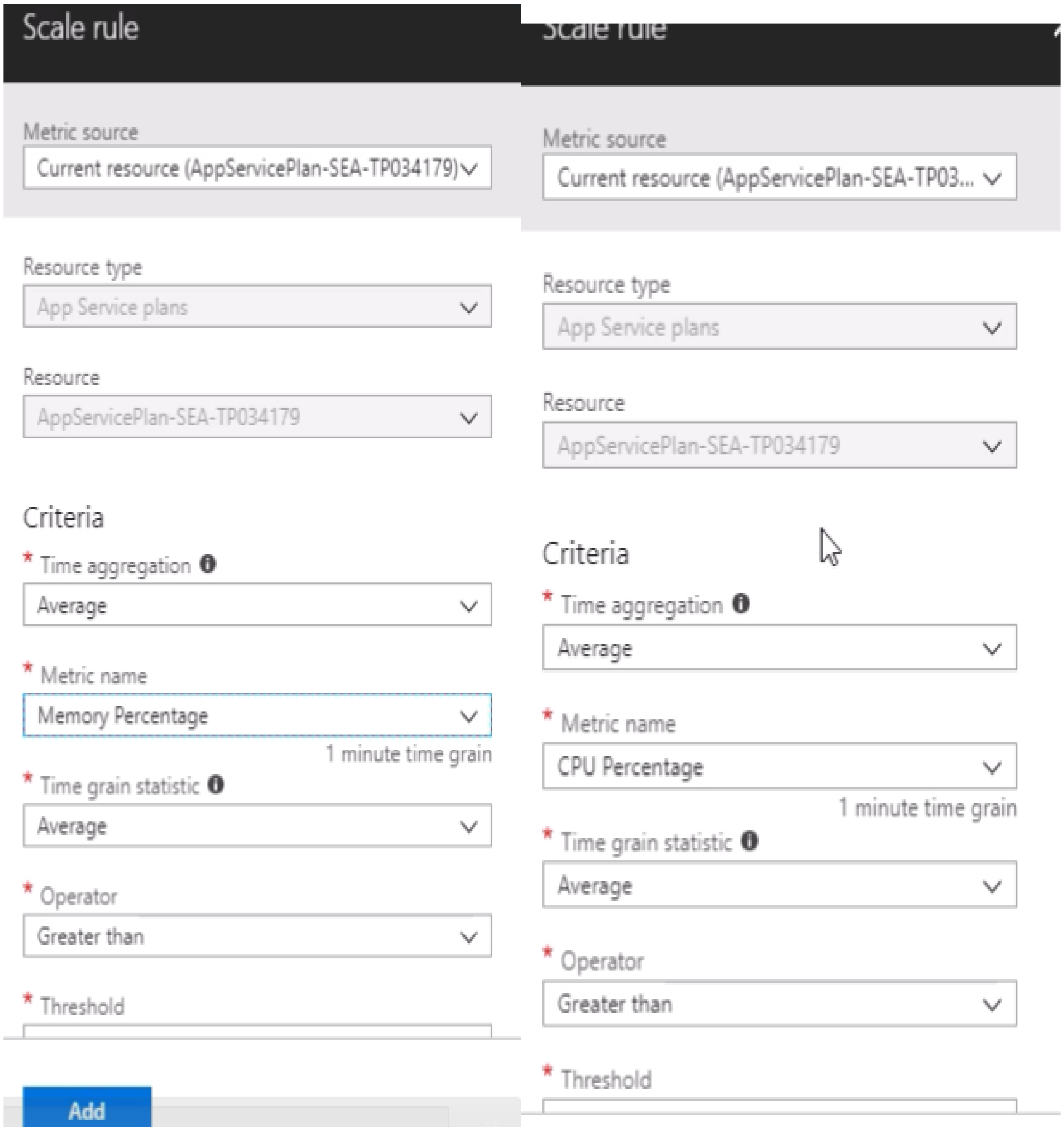
Rules

Scale out			
When	AppServicePlan-S...	(Average) CpuPercentage > 70	Increase instance count by 1
Or	AppServicePlan-S...	(Average) MemoryPercentage > 70	Increase instance count by 1

Figure 40: Setup Web App Scale Out

The purpose of setting up Auto Scaling for Maersk Line project is because it can able to scale up and scale out the application service plan automatically based on the criteria of the users. There are two options for the Scale mode in “Scale Out (App Service Plan” which are “Scale based on a metric” or “Scale to a specific instance count”. For Maersk Line project, it is required a “Scale based on a metric”.

4.7.2 Setup Scale Rule for Auto Scaling Plan



Scale rule

Metric source
Current resource (AppServicePlan-SEA-TP034179)

Resource type
App Service plans

Resource
AppServicePlan-SEA-TP034179

Criteria

* Time aggregation
Average

* Metric name
Memory Percentage
1 minute time grain

* Time grain statistic
Average

* Operator
Greater than

* Threshold

Scale rule

Metric source
Current resource (AppServicePlan-SEA-TP03...

Resource type
App Service plans

Resource
AppServicePlan-SEA-TP034179

Criteria

* Time aggregation
Average

* Metric name
CPU Percentage
1 minute time grain

* Time grain statistic
Average

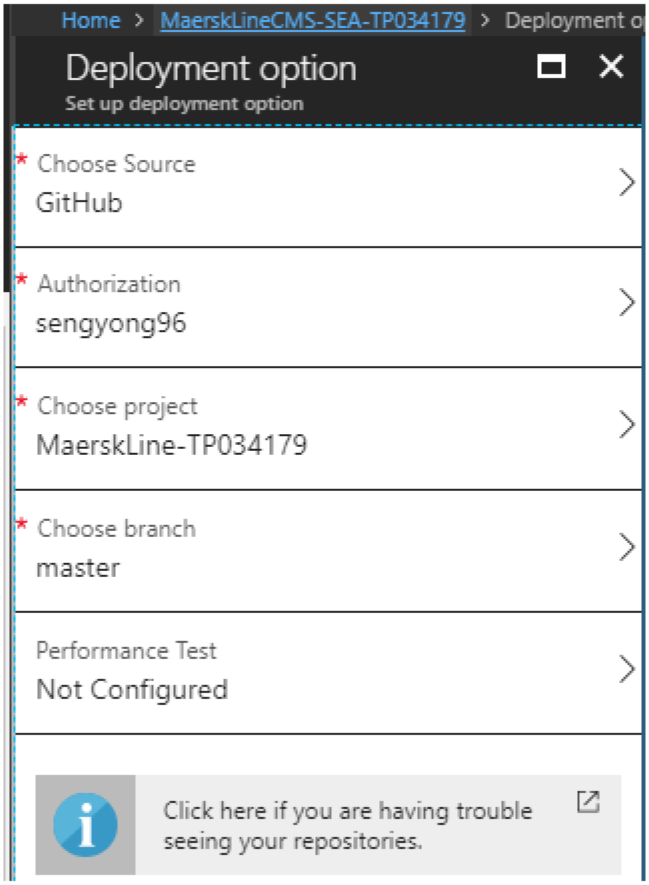
* Operator
Greater than

* Threshold

Figure 41: Scale Rules for Auto Scaling

The scale rules for the auto scaling are based on the criteria of following “Metric name”. There are different types of Metric name for the auto scaling which includes “Memory Percentage” and “CPU Percentage”. These two are the criteria for the Maersk Line auto scaling metric.

4.8 Link to GitHub



Deployment option
Set up deployment option

* Choose Source
GitHub

* Authorization
sengyong96

* Choose project
MaerskLine-TP034179

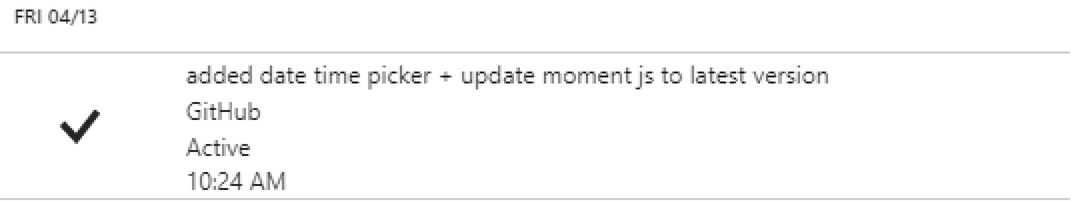
* Choose branch
master

Performance Test
Not Configured

Click here if you are having trouble seeing your repositories.

Figure 42: GitHub on Azure

By connecting GitHub to Azure, select Web App Service that users wanted to connect to GitHub. After that, select “Deployment Option”, then select “GitHub”, and enter the credentials of the GitHub and authorize it. After that, choose the project of the Maersk Line project in GitHub, and click Add.


FRI 04/13

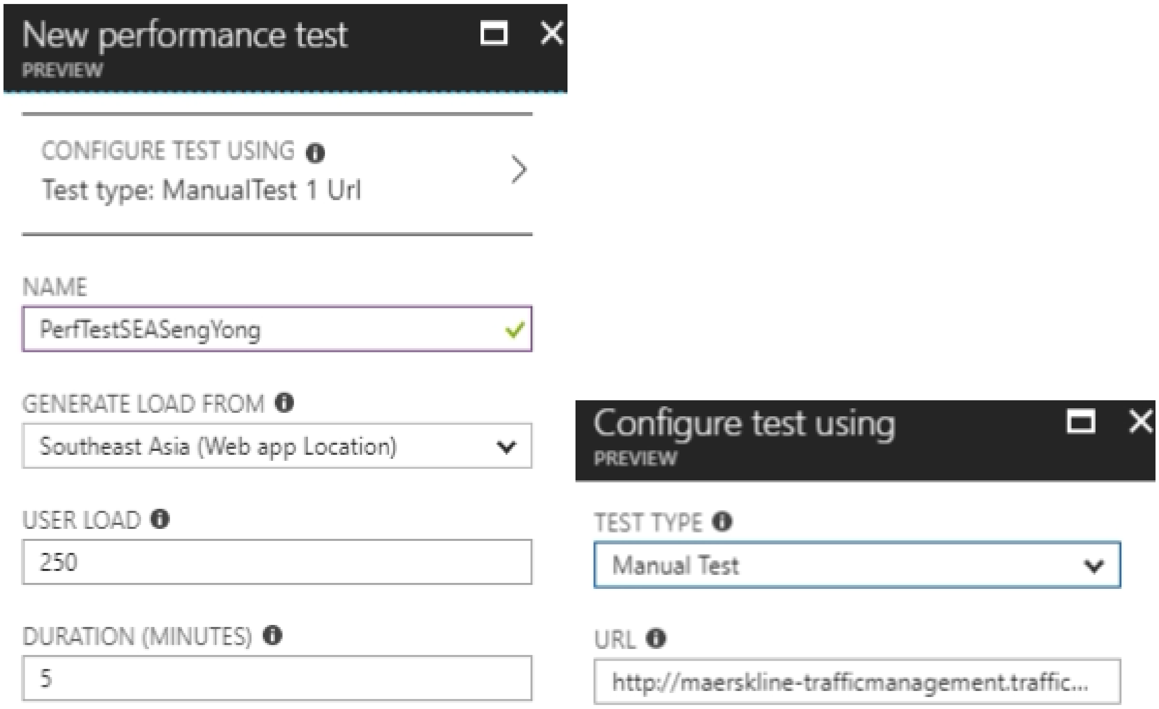
✓ added date time picker + update moment js to latest version
GitHub
Active
10:24 AM

Figure 43: Done Linking to GitHub

GitHub is now connected to Azure as shown in the figure above.

5.0 Testing

5.1 Performance Testing



New performance test
PREVIEW

CONFIGURE TEST USING
Test type: ManualTest 1 Url

NAME
PerfTestSEASengYong

GENERATE LOAD FROM
Southeast Asia (Web app Location)

USER LOAD
250

DURATION (MINUTES)
5

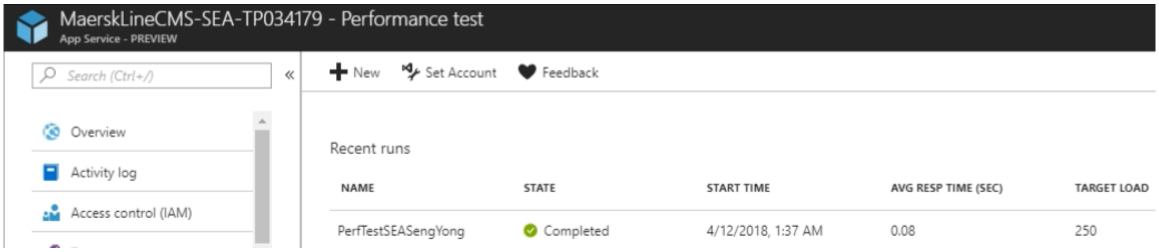
Configure test using
PREVIEW

TEST TYPE
Manual Test

URL
http://maerskline-trafficmanagement.traffic...

Figure 44: Performance Testing for Maersk Line

There is performance testing for Maersk Line project as it can Check your web app's performance before you launch it or deploy updates to production (Docs.microsoft.com, 2018). By doing so, user can select the Web App Service that they want to perform performance test to it, then they will redirect to a new page where they can find an option “Performance Test”. After that, Click “New” to perform a new performance testing for the MaerskLine project. The Test type will be using ManualTest 1 Url, and the URL will be the traffic manager URL that has been done earlier. The performance test has named as PerfTestSEASengYong for testing the region of SEA and have a user load of 250 within the duration of 5 minutes. The performance test usually will take 10-20 minutes to complete as initially the status of the testing will be “Queued”, then “In Progress”, then “Completed” as shown in the figure below.

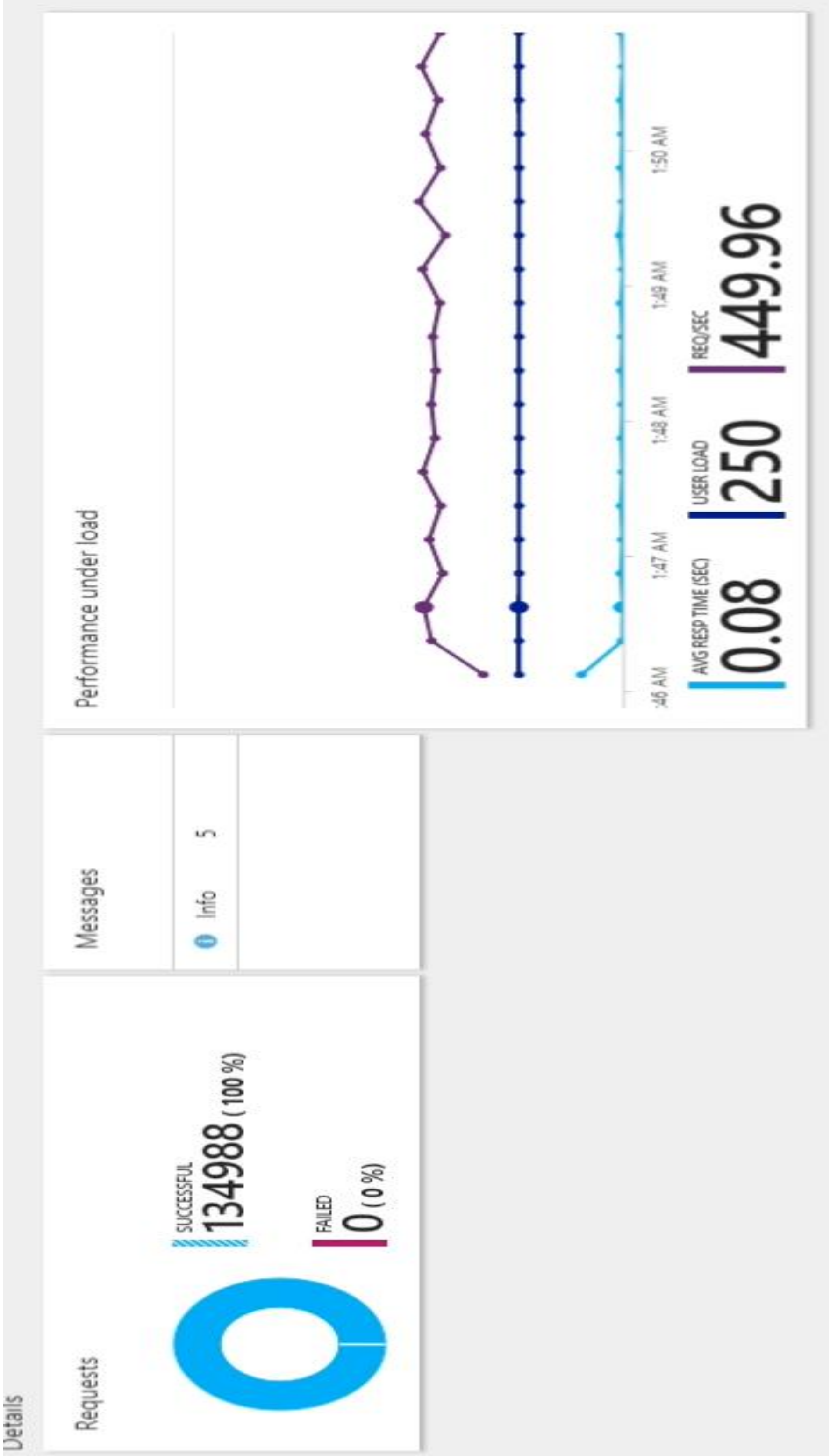


Recent runs

NAME	STATE	START TIME	AVG RESP TIME (SEC)	TARGET LOAD
PerfTestSEASengYong	✓ Completed	4/12/2018, 1:37 AM	0.08	250

Figure 45: Completed Performance Test

The result of the performance test for the Maersk Line Web App will be shown at the figure below as the result is showing in a pie chart and in a graph. The pie chart is showing the successful requests and failed requests. There are 134988 test requests are performed and none of it are failed, there is 100% successful and 0% of failure. It has shown that the system is very reliable and has the high availability for users. Moreover, the performance test has included the result of the average response time per second (AVG RESP TIME (SEC)), user load, requests per second, where the results are shown accordingly, 0.08 for AVG RESP TIME, 250 for the User Load, 449.96 for the requests per second. As for the graph showing in the performance test result, it can see the performance under load in very detailed by just moving the cursor within the graph and it can see the response time, and request per second in a specific time frame.



Details

Requests

SUCCESSFUL 134988 (100 %)

FAILED 0 (0 %)

Messages

Info 5

Performance under load

AVG RESP TIME (SEC) 0.08

USER LOAD 250

REQ/SEC 449.96

Figure 46: Results of the Performance Test for Maersk Line Web App

5.2 Unit Testing

5.2.1 Login

TestCaseID	Test Case Description	Expected Result	Actual Result
T01	1a. Enter Correct Email 1b. Enter Correct Password 2a. Enter Correct Email 2b. Enter Incorrect Password 3a. Enter Incorrect Email 3b. Enter Incorrect Password 4a. Enter Incorrect Email 4b. Enter Correct Password	1. Login Successfully 2. Login Failed 3. Login Failed 4. Login Failed	1. Login Successfully 2. Login Failed 3. Login Failed 4. Login Failed

Table 8: Unit Testing for Login

5.2.2 Register Agent

TestCaseID	Test Case Description	Expected Result	Actual Result
T02	1a. Enter Correct Email 1b. Enter Correct Password 2a. Enter Correct Email 2b. Enter Incorrect Password 3a. Enter an existing Email 3b. Enter Correct Password	1. Register Successfully 2. Register Failed 3. Register Failed	1. Register Successfully 2. Register Failed 3. Register Failed

Table 9: Unit Testing for Register Agent

5.2.3 Manage Schedule

TestCaseID	Test Case Description	Expected Result	Actual Result
Create Schedule			
T03-1	1. Enter Correct Schedule information 2. Enter Incorrect Schedule information	1. Create schedule successfully. 2. Create schedule failed.	1. Create schedule successfully. 2. Create schedule failed.
Edit Schedule			
T03-2	1. Edit schedule with Correct Schedule information 2. Edit schedule with Incorrect Schedule information 3. Edit schedule that one or more ships has already assigned with it	1. Edit schedule successfully. 2. Edit schedule failed. 3. Edit schedule failed.	1. Edit schedule successfully. 2. Edit schedule failed. 3. Edit schedule failed.
Delete Schedule			
T03-3	1. Delete schedule with no ship has assigned with it. 2. Delete schedule with one or more ships has assigned with it.	1. Delete schedule successfully. 2. Delete schedule failed.	1. Delete schedule successfully. 2. Delete schedule failed.
View Schedule			
T03-4	1. Select schedule to view	1. View Schedule successfully	1. View Schedule successfully

Table 10: Unit Testing for Manage Schedule

5.2.4 Manage Ship

TestCaseID	Test Case Description	Expected Result	Actual Result
	Create Ship		
T04-1	1. Enter Correct Ship information 2. Enter Incorrect Ship information	1. Create Ship successfully. 2. Create Ship failed.	1. Create Ship successfully. 2. Create Ship failed.
	Edit Ship		
T04-2	1. Edit Ship with Correct Ship information 2. Edit Ship with Incorrect Ship information 3. Edit Ship that one or more schedule has already assigned with it	1. Edit Ship successfully. 2. Edit Ship failed. 3. Edit Ship failed.	1. Edit Ship successfully. 2. Edit Ship failed. 3. Edit Ship failed.
	Delete Ship		
T04-3	1. Delete Ship with no schedule has assigned with it. 2. Delete Ship with one or more schedules has assigned with it.	1. Delete Ship successfully. 2. Delete Ship failed.	1. Delete Ship successfully. 2. Delete Ship failed.
	View Ship		
T04-4	1. Select Ship to view	1. View Ship successfully	1. View Ship successfully

Table 11: Unit Testing for Manage Ship

5.2.5 Manage Customer

TestCaseID	Test Case Description	Expected Result	Actual Result
	Create Customer		
T05-1	1. Enter Correct Customer information 2. Enter Incorrect Customer information	1. Create Customer successfully. 2. Create Customer failed.	1. Create Customer successfully. 2. Create Customer failed.
	Edit Customer		
T05-2	1. Edit Ship with Correct Customer information	1. Edit Customer successfully. 2. Edit Customer failed.	1. Edit Customer successfully. 2. Edit Customer failed.

	2. Edit Customer with Incorrect Customer information		
Delete Customer			
T05-3	1. Select Customer to delete	1. Delete Customer successfully.	1. Delete Customer successfully.
View Customer			
T05-4	1. Select Customer to view	1. View Customer successfully	1. View Customer successfully

Table 12: Unit Testing for Manage Customer

5.2.6 Manage Booking

TestCaseID	Test Case Description	Expected Result	Actual Result
Create Booking			
T06-1	1. Enter Correct Booking information 2. Enter Incorrect Booking information	1. Create Booking successfully. 2. Create Booking failed.	1. Create Booking successfully. 2. Create Booking failed.
View Booking			
T06-2	1. View Booking	1. View booking successfully	1. View booking successfully

Table 13: Unit Testing for Manage Booking

6.0 Conclusion

In conclusion, the cloud-based application, Maersk Line application allows users to create ship, schedule, customer, make booking and add container. The Maersk Line web application has been developed completely and published it to Microsoft Azure cloud platform. It has provided the best solution for Maersk Line company to meet all the requirement that the company needs. Maersk Line application can now reduce overall supply chain costs and has an efficient way to manage logistics. Besides, Microsoft Azure cloud platform allows developer to deploy the web application in a very easy way. It is easy to use. Developer has implemented traffic manager, geo-replication, failover policy, auto scaling, and performance test for Maersk Line web application.

Lastly, developer has learnt a lot of knowledges during the development of this project for Maersk Line company. Developer can highly ensure that he is able to deploy any web application to Azure cloud platform in just less than 1 hour in future.

7.0 References

Russell Smith (2015). What are Microsoft Azure Resource Groups?. [online] Petri. Available at: https://www.petri.com/what-are-microsoft-azure-resource-groups [Accessed 12 Apr. 2018].

Dwivedi, K., Madureira, J. and Tuliani, J. (2017). What is Traffic Manager. [online] Docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview [Accessed 12 Apr. 2018].

Docs.microsoft.com. (2018). Test your Azure web app performance under load from the Azure portal. [online] Available at: https://docs.microsoft.com/en-us/vsts/load-test/app-service-web-app-performance-test?view=vsts [Accessed 12 Apr. 2018].

Lin, C., Borges, B., Fowler, C. and Gangula, R. (2017). Web Apps overview. [online] Docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/azure/app-service/app-service-web-overview [Accessed 12 Apr. 2018].

8.0 Appendix

1. GitHub URL: https://github.com/sengyong96/MaerskLine-TP034179.git

2. Azure Web Application URL:

- http://maersklinecms-sea-tp034179.azurewebsites.net/ (SEA Region)
- http://maersklinecms-ncus-tp034179.azurewebsites.net/ (NCUS Region)
- http://maerskline-trafficmanagement.trafficmanager.net (Traffic Manager)

3. Azure and Web App System Demonstration URL:

- https://tinyurl.com/y76p2xha (tinyurl)
- https://web.microsoftstream.com/video/df3e1da5-ad77-4164-ace3-20db1a1e5bdc?search=TP034179 (original URL)

4. User login credentials

Admin Username: admin@maersk.com

Admin Password: Admin123.

Agent Username: agent@maersk.com

Agent Password: Agent123.