PT 4

AMARJEET BRIJNANDAN SINGH - A0150157M  
  
CHENG HAO - A0006467E  
  
GONG SHENGLIANG - A0112429M  
  
HU RENWEN - A0150352U  
  
SMITA RANI BISOYI - A0150422X

Cloud computing

Active Learning on the Cloud Project Report

Team member Contributions

|  |  |
| --- | --- |
| Team member | Contributions |
| AMARJEET BRIJNANDAN SINGH | * Explore and setup Azure Docker Service * Explore and setup Azure Storage Account |
| CHENG HAO | * Modify application to utilize Azure blob storage * Explore and setup Traffic Manager * Explore and setup Application Insights * Explore and setup shared Dashboards |
| GONG SHENGLIANG | * Deploy Web App * Deploy Azure DB * Explore and setup Web App auto scaling |
| HU RENWEN | * Explore and setup Azure DB Geo replication * Explore and setup alert for Web App Service Plan and Azure DB recourse usage alert |
| SMITA RANI BISOYI | * Explore and setup Azure Logic App * Explore and integrated SendGrid email API |

Some Notes:

The following were not selected for the presentation, reason being:

* Docker runs on VM which is too costly for the Geo redundancy
* SendGrid free account has limited service time, and it is not really unique cloud feature

Contents

[Background 3](#_Toc486770852)

[Architecture 4](#_Toc486770853)

[Setup Journal 5](#_Toc486770854)

[1. App Service 5](#_Toc486770855)

[a. App Service Plan with Geo Redundancy 5](#_Toc486770856)

[b. Web App 5](#_Toc486770857)

[c. Deploy application 6](#_Toc486770858)

[d. Auto scaling 7](#_Toc486770859)

[e. Usage Alert 9](#_Toc486770860)

[2. Azure DB 9](#_Toc486770861)

[a. Create DB schema 9](#_Toc486770862)

[b. Geo Redundancy 9](#_Toc486770863)

[c. Usage Alert 9](#_Toc486770864)

[3. Azure Storage Account 9](#_Toc486770865)

[a. Create Azure Storage Account 9](#_Toc486770866)

[b. Code change to utilize Blob Storage 9](#_Toc486770867)

[4. Application Insights 9](#_Toc486770868)

[a. Create Application Insights App 9](#_Toc486770869)

[b. Code change to trigger page tracking 9](#_Toc486770870)

[5. Traffic Manager 9](#_Toc486770871)

[a. Create Traffic Manager Profile 9](#_Toc486770872)

[b. Mode Configuration 9](#_Toc486770873)

[c. Add End Point 9](#_Toc486770874)

[6. Cloud Performance Test 9](#_Toc486770875)

[a. Visual Studio Performance Test Project 9](#_Toc486770876)

[7. Azure Logic App 10](#_Toc486770877)

[a. Data source preparation 10](#_Toc486770878)

[b. Azure Logic App Creation 10](#_Toc486770879)

[8. Geo redundancy failover simulation 13](#_Toc486770880)

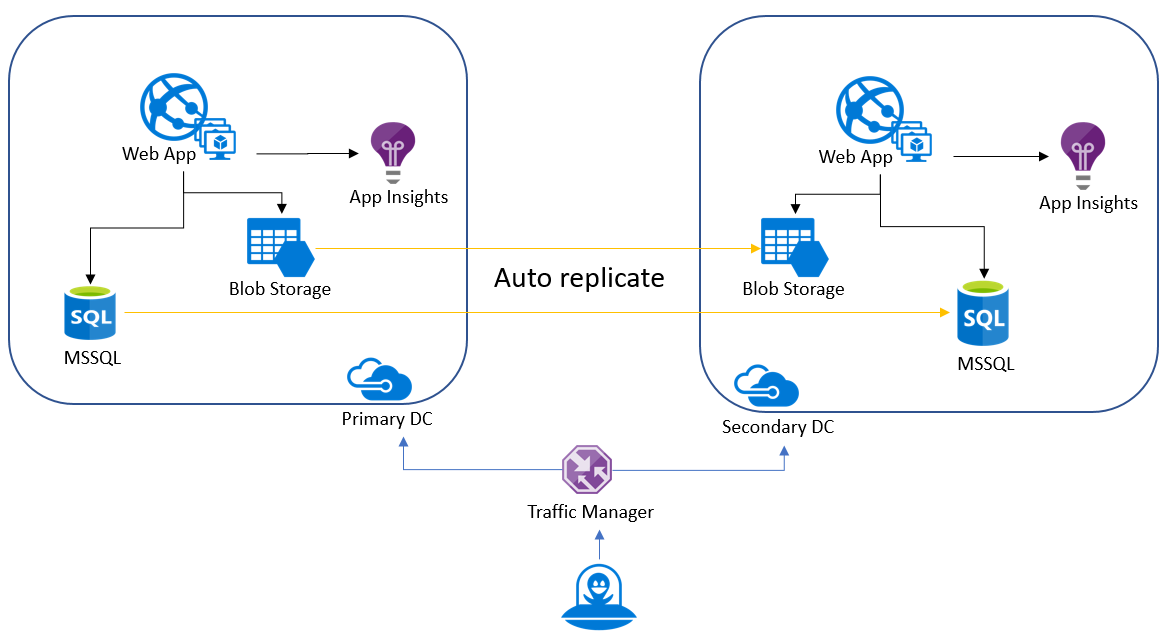
[9. IIS Remote Management 13](#_Toc486770881)

[Lesson learned and challenges faced 13](#_Toc486770882)

# **Background**

* Active Learning Institute had launched an e-learning system in 2016. It was hosted on Microsoft Azure VM (IaaS) with localhost SQL database.
* There were high demands of the registration and course online browsing, which caused poor performance of the system.
* The IT department head of the institute is looking at optimizing the program of the system as well as a better hosting solution to achieve quick and flexible resource scaling, and yet at a reasonable cost.
* After a few internal meetings, the IT department head decided to migrate the system to Microsoft Azure
  + Web App (PaaS)
  + Azure MSSQL
  + Blob Storage
  + Traffic Manager

# **Architecture**



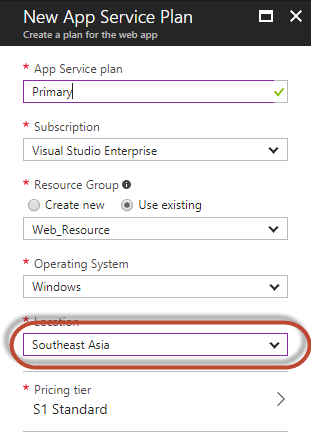
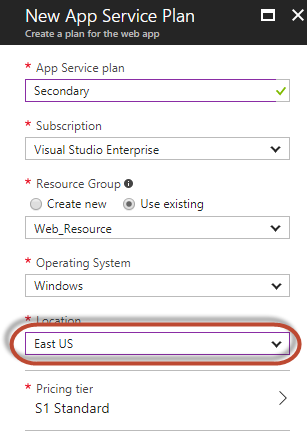
# **Setup Journal**

## App Service

### App Service Plan with Geo Redundancy

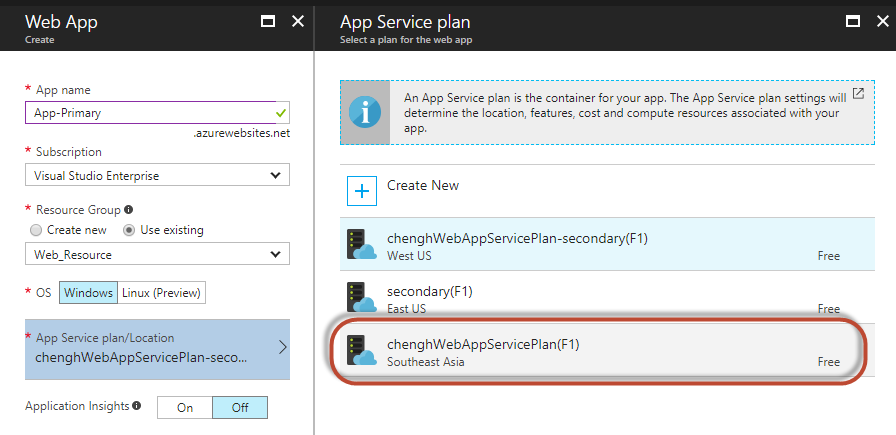
Create 2 App Service Plans in different zones

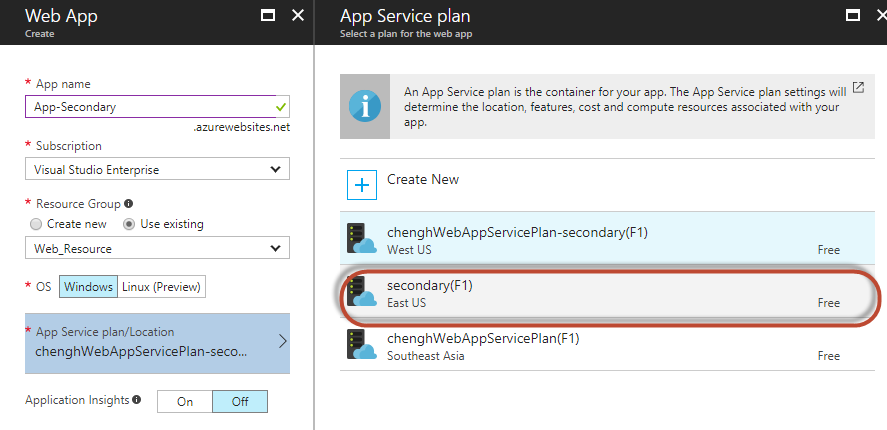
Consideration: Choose the same datacentre as database server or a datacentre near the database server datacentre, to archive low latency.

### Web App

Create 2 Web Apps, tagged under the respective App Service Plan





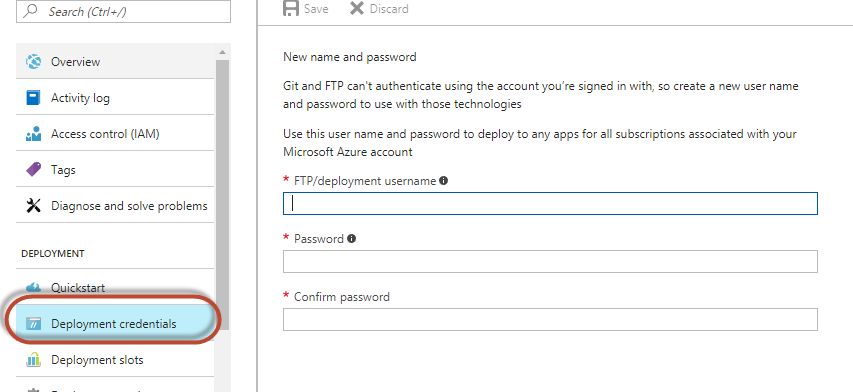
### Deploy application

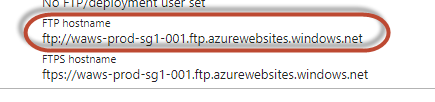
We have explored and learned 2 ways to deploy the runtime files to Azure Web App.

Each Web App will require one deployment. Hence 2 zone Geo-redundancy will require 2 separate manual deployments.

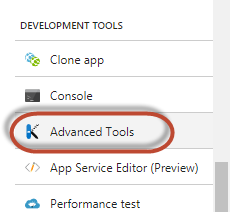
1. Via FTP

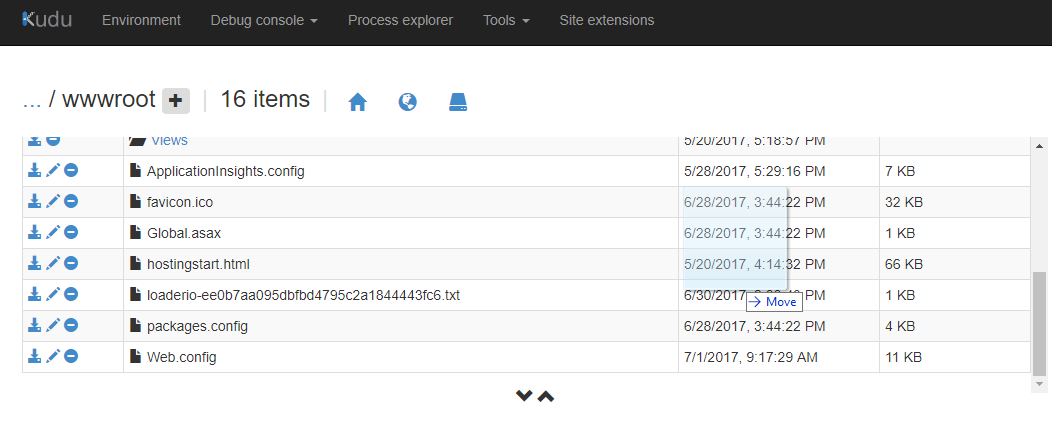
Configure the FTP account followed by accessing the FTP path under overview





1. Via Advanced Tools (Drag & Drop Web Deploy)

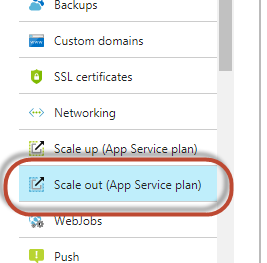




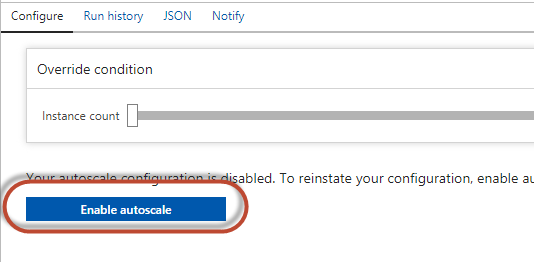
### Auto scaling

Note: Auto scaling requires min standard tier

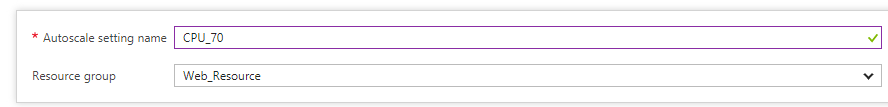
Go the Web App => Sale out.



Enable autoscale

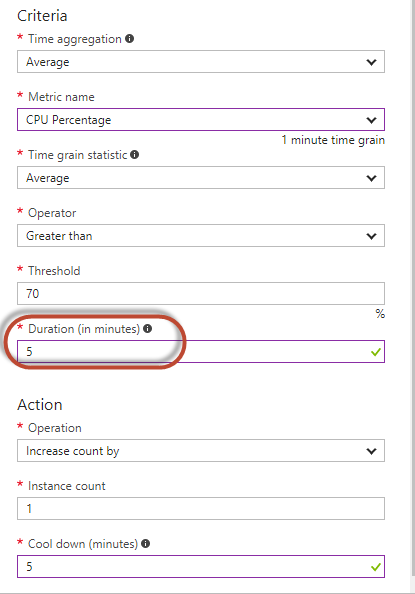


Specify autoscale configuration name



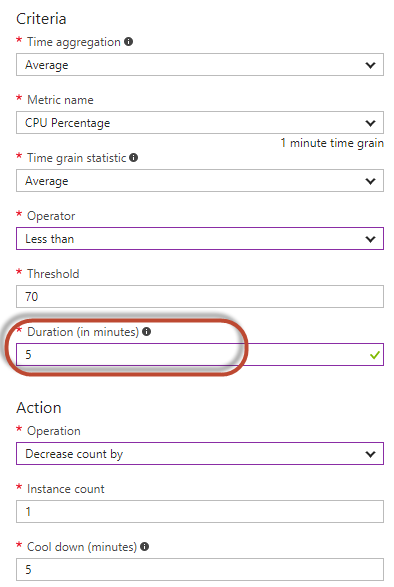
Create auto scale out rule

Set a shorter duration for a quicker auto scale out action, to minimize the impact by resource overload



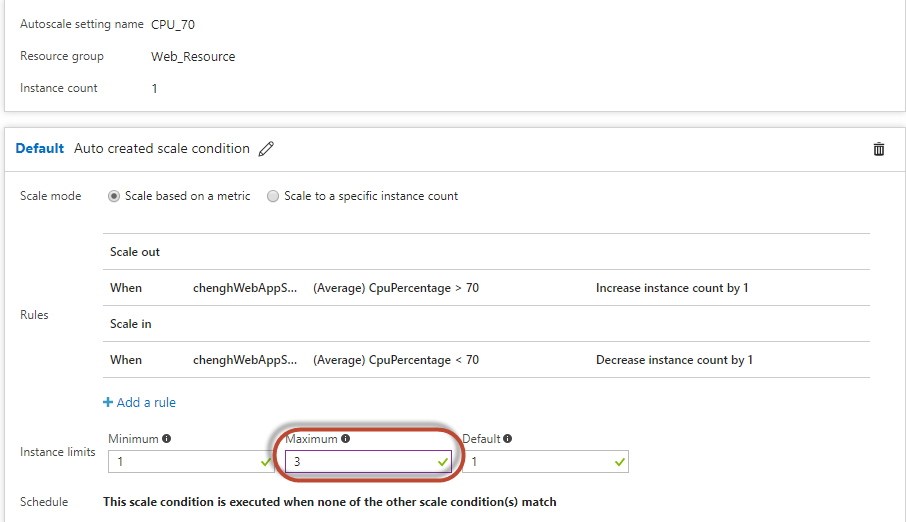
Create auto scale in rule

Set a shorter duration for a quicker auto scale in action, to archive more cost-effetiveness

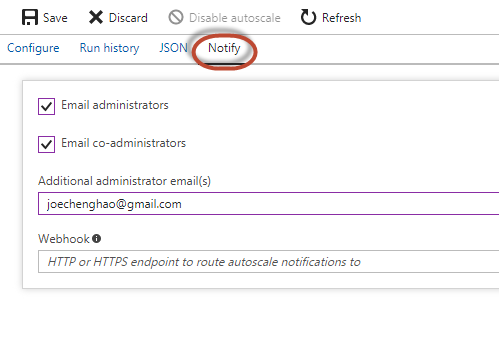


Overview

Change maximum provisioned instances to be more than 1

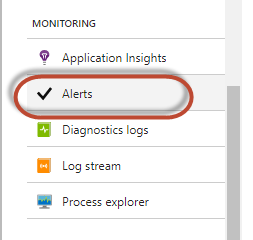


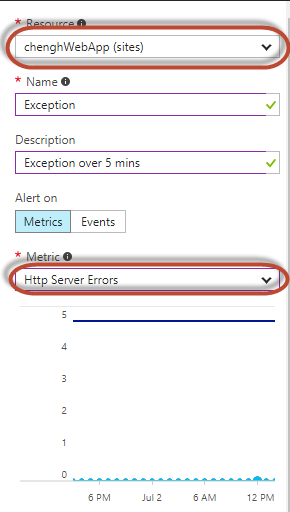
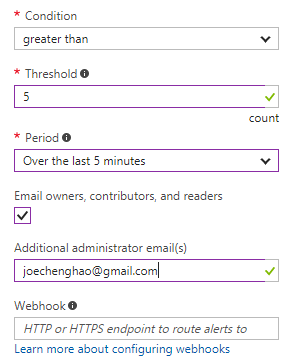
Autoscale alert



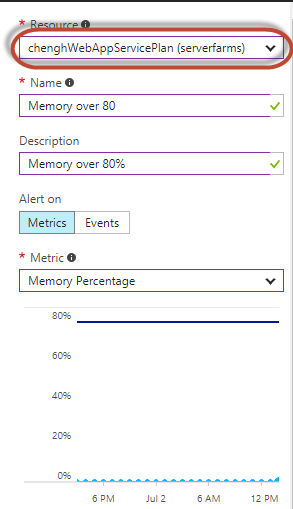
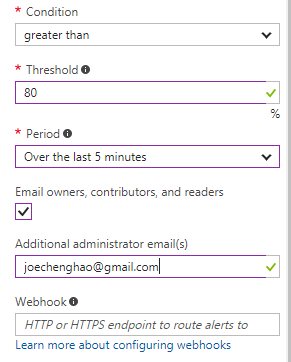
### Web App Exception Alert

Go to monitoring => alert



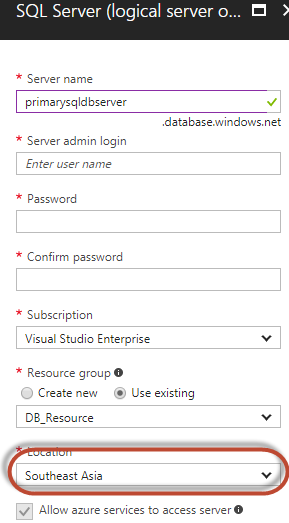
### App Service Plan Usage Alert

## Azure SQL database

### Create Azure SQL database Server

Create 1 SQL database server in the primary datacentre, to be in the same datacentre as the primary App Service Plan or nearest.



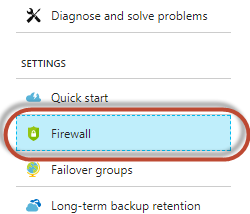
### Create Azure SQL database

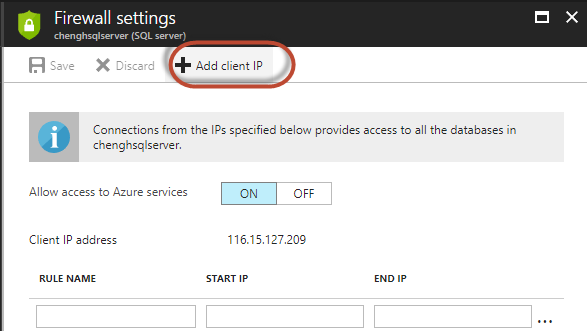
We explored 2 ways to create Azure SQL database

Since we will setup Geo Replication, we only need to setup the primary SQL database.

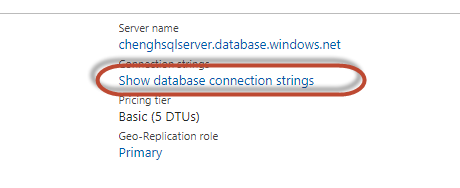
1. Via MSSQL management studio (SMSS)

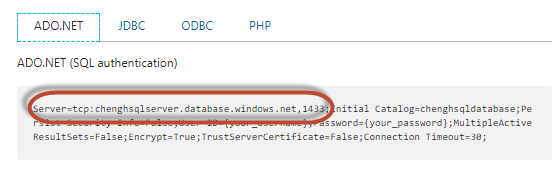
Prerequisite: Client IP is whitelisted in Azure SQL database server firewall



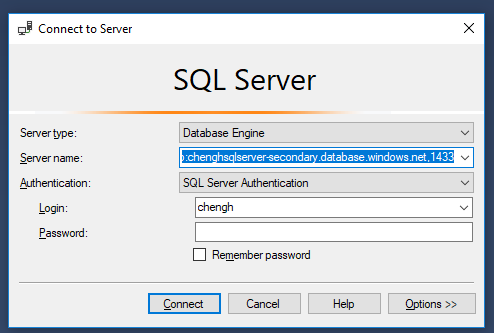


Get the connection string from SQL database

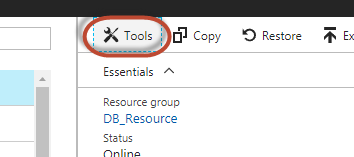




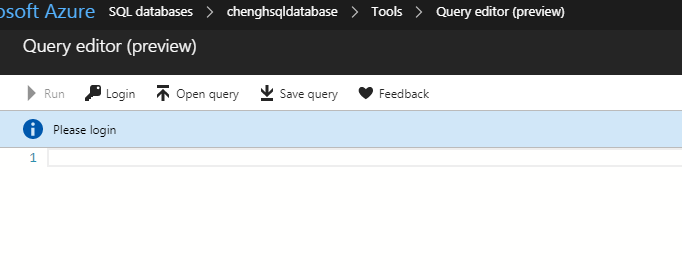
Connect via SMSS as per normal



1. Via query editor

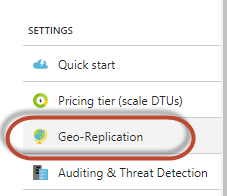


Run the full setup SQL script in the editor

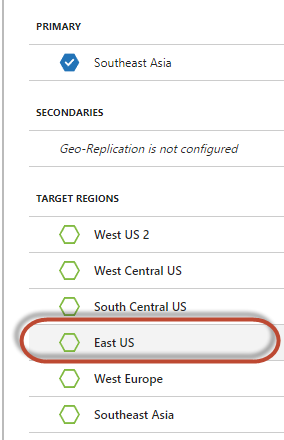


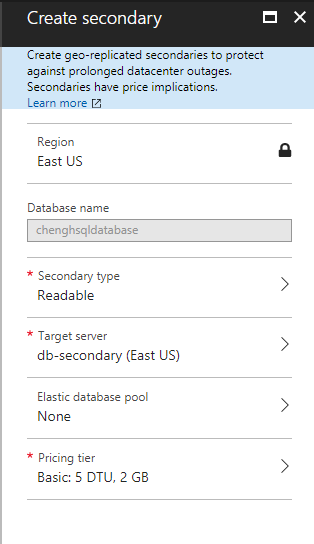
### Geo Replication

Go to the primary SQL database => Geo-Replication

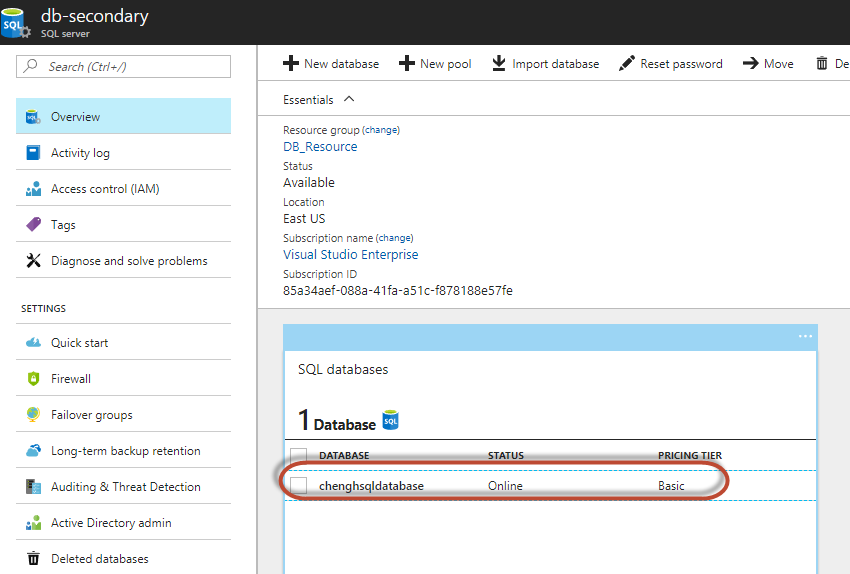


Choose a region the same as the secondary App Service Plan or nearest





A SQL database with the same name is created in the target replication SQL server.



### Application database connection string change

For the primary and secondary Web App to connect to its respective SQL database, database connection string has to be changed, however only the SQL server address.

E.g.

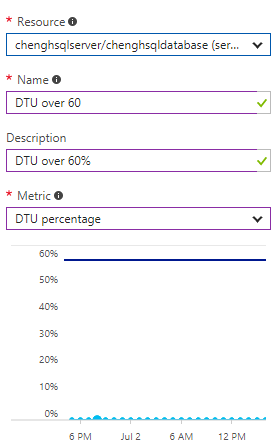
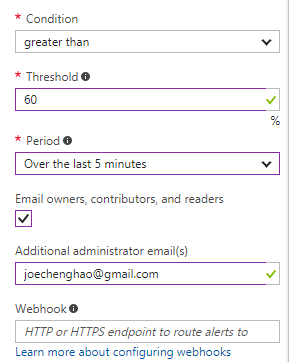
Primary: tcp:chenghsqlserver.database.windows.net,1433

Secondary: tcp:chenghsqlserver-secondary.database.windows.net,1433

### Usage Alert

Go to Monitoring => Alert rules to add alert

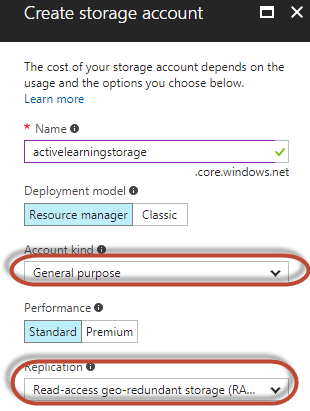


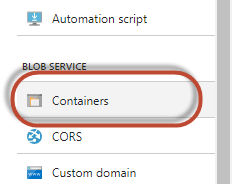
## Azure Storage Account

### Create Azure Storage Account

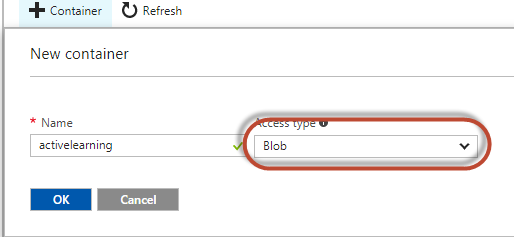
Create storage account



Create storage container



Choose blob for access type for the application to be able to access by the address of the file



### Code change to utilize Blob Storage

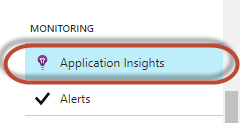
Some amount of codes need to be written in order to upload the contents to Azure storage, download content from and delete from Azure storage. Please refer to the solution codes => ActiveLearning.Business => Common => Util.cs

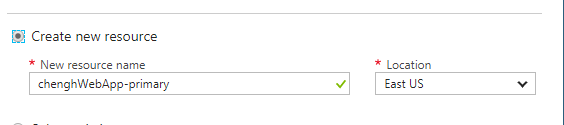
## Application Insights

### Create Application Insights App

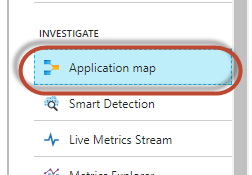
We will need to configure 2 times, for both the primary and secondary Web Apps.

Go to Web App => monitoring => Application Insights

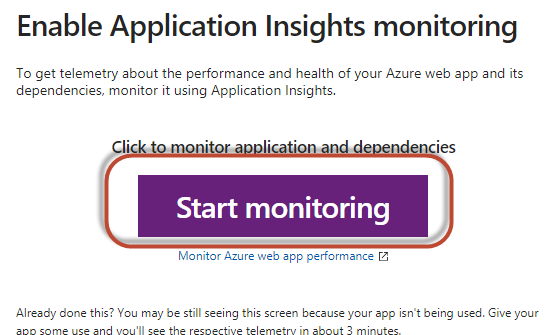




Go to Application Insights => Investigate => Application map

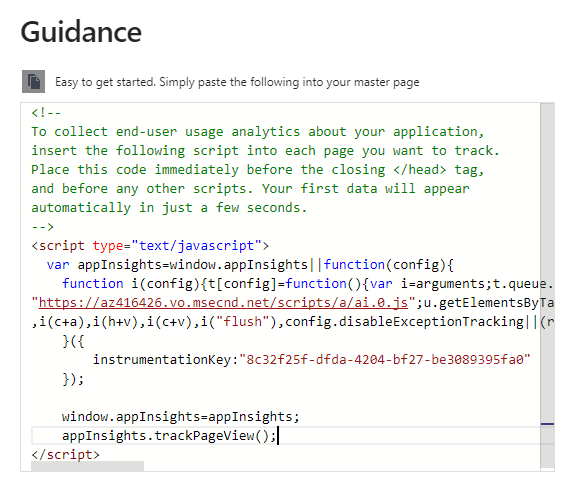


Start the monitoring.



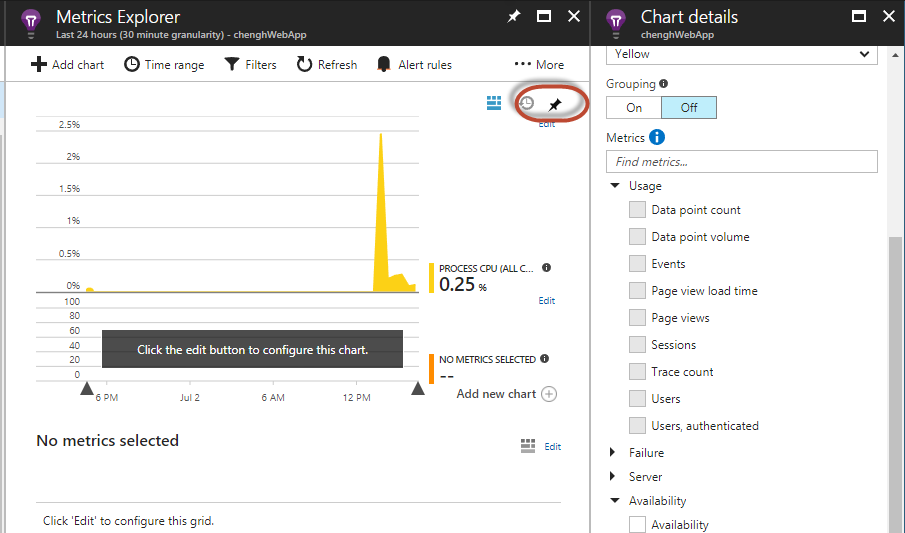
### Code change to trigger client page tracking

Click on the Client box in the application map, and copy the JavaScript code to your master page or any standalone frontend web pages.



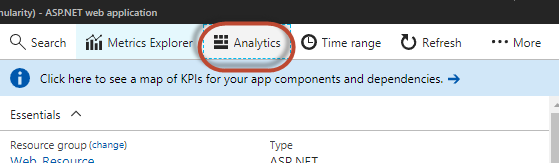
### Metrics monitoring

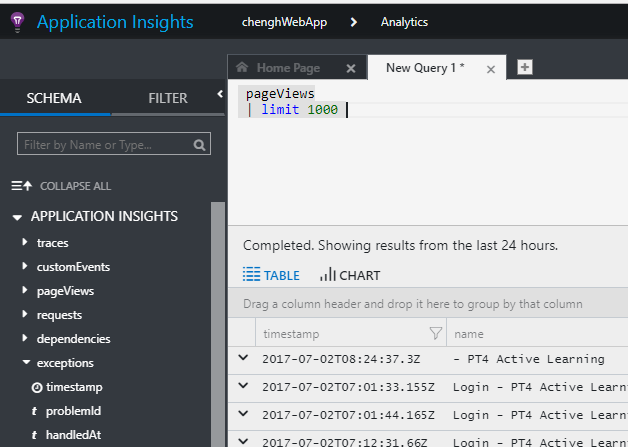
Go to the metric explorer to customize the metric, and Pin to the dashboard



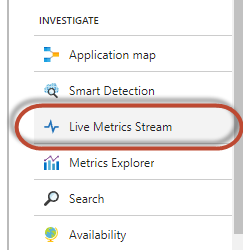
### Analytics tool

The analytics tool provide more flexibility when building customized query in the metric data.

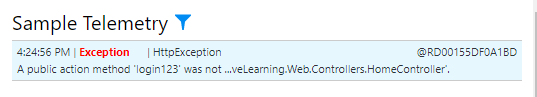


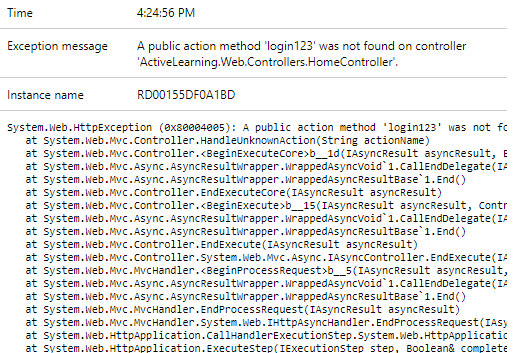


### Live stream



Realtime requests and resource usage are shown, as well as exceptions. It is very useful during a live event. Able to provide real time performance of the application for the support team.



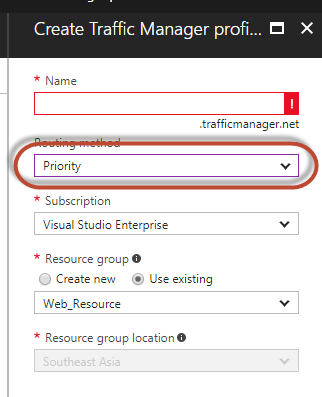


## Traffic Manager

### Create Traffic Manager Profile

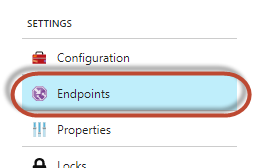
Note: All the Web Apps must be min standard tier in order to be included in Traffic Manager monitoring and traffic forwarding.

Choose Priority method for Geo Redundancy

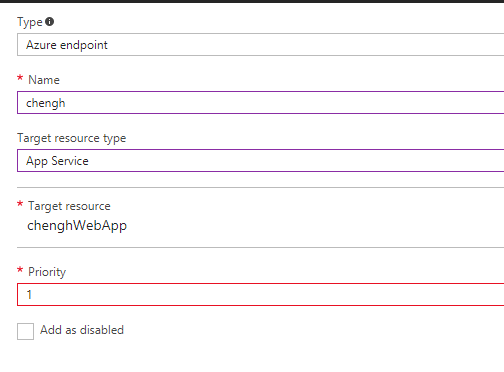


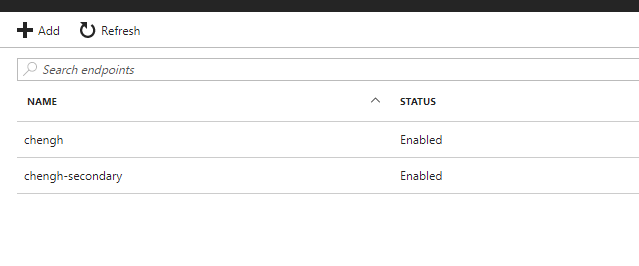
### Add End Point

Add both Web Apps in the end point



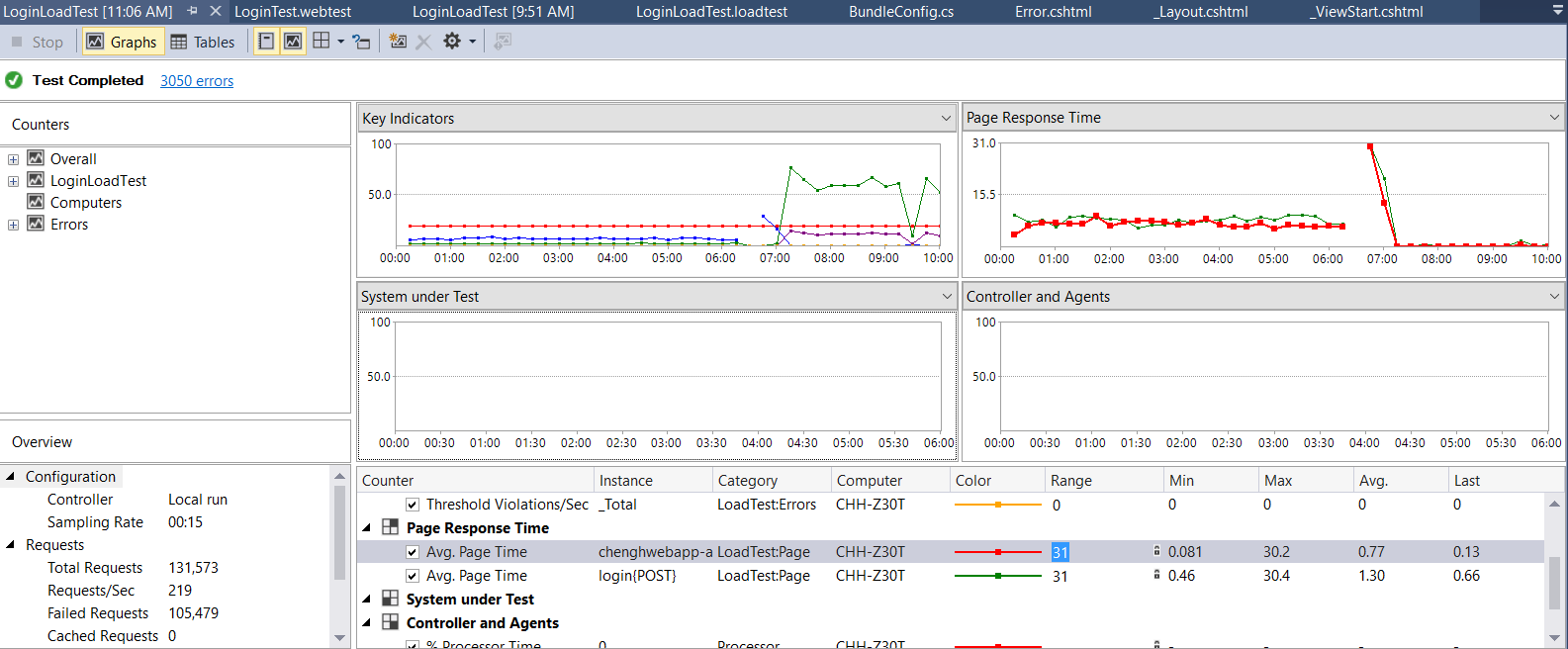
Smaller priority number denotes higher priority. Hence primary is 1, secondary is 2.





## Cloud Performance Test

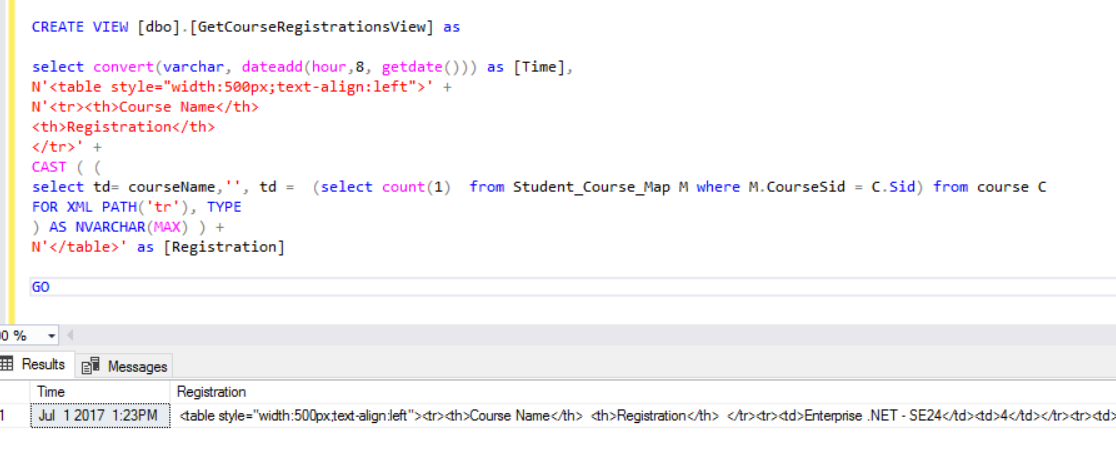
### Visual Studio Performance Test Project



## Azure Logic App

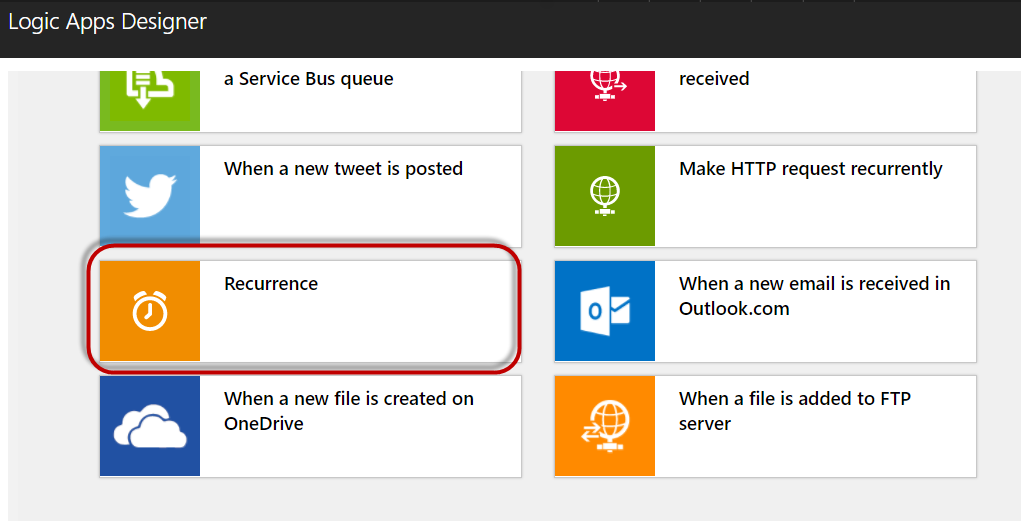
### Data source preparation

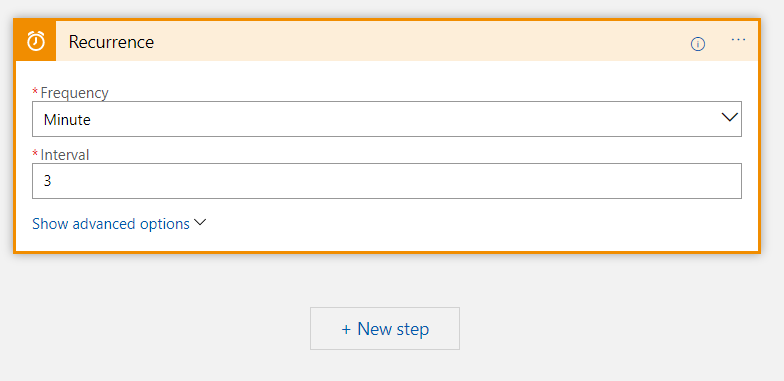
MSSQL View to construct the HTML contents



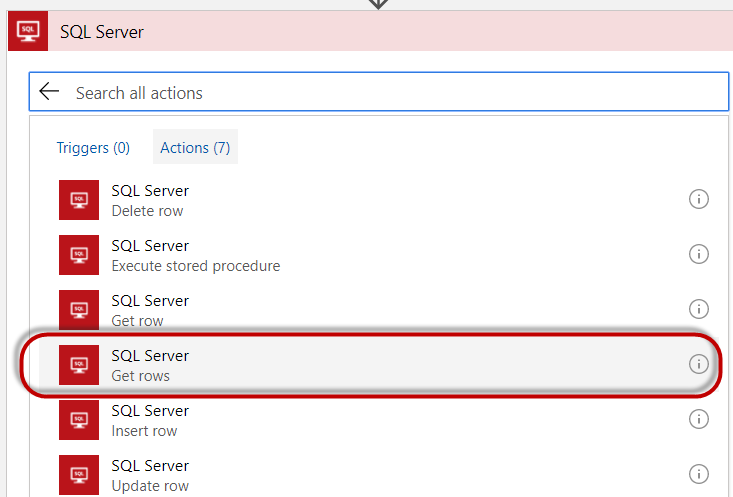
### Azure Logic App Creation

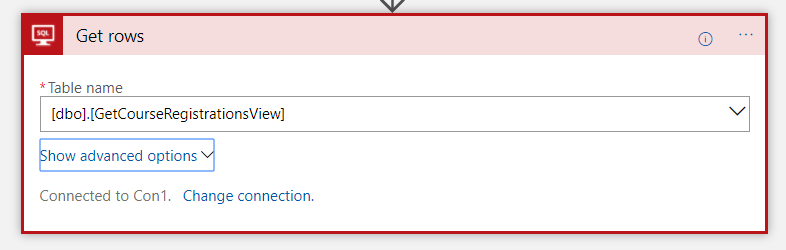
1. Setup trigger



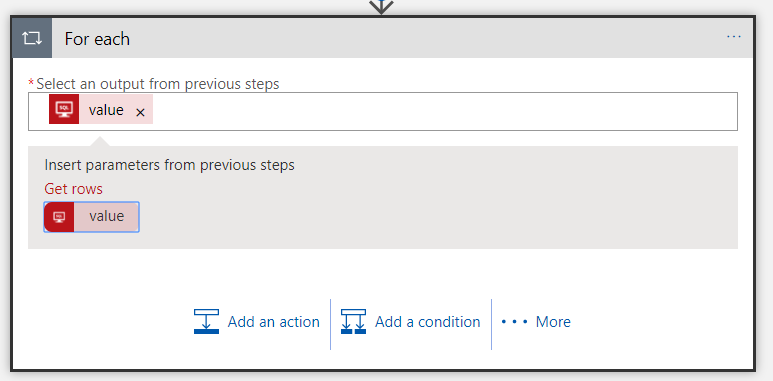


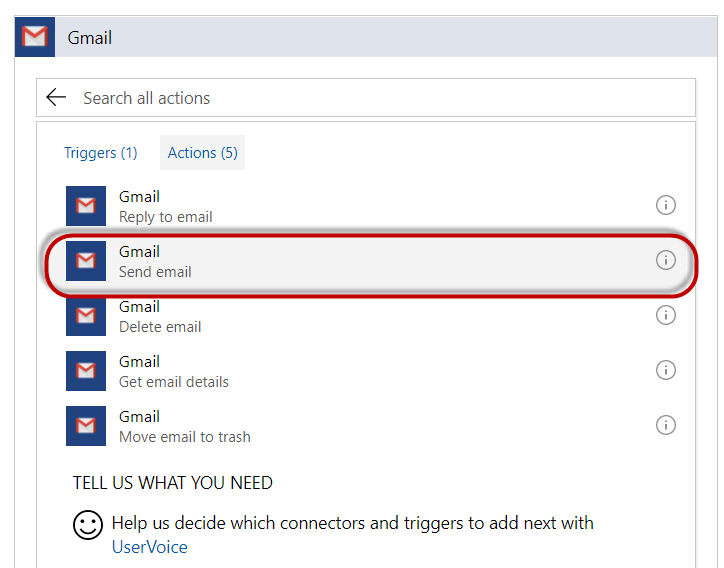
1. Retrieve data

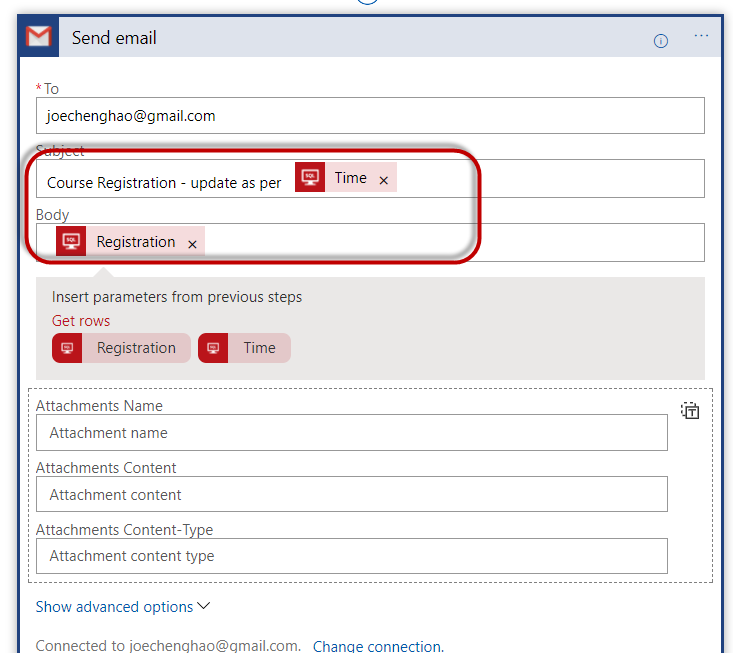


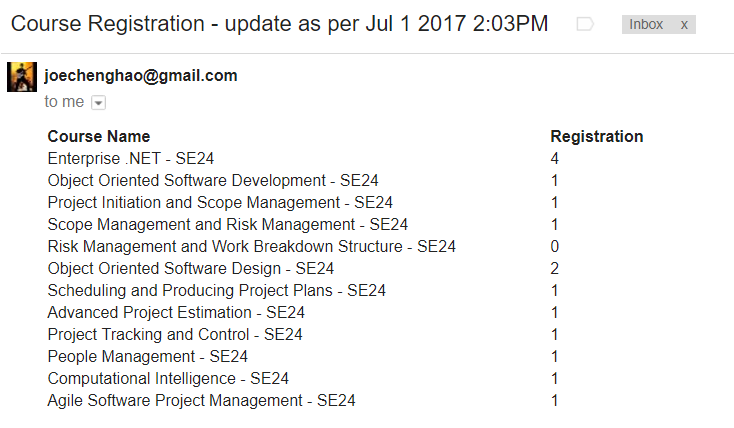


1. Process data and send email









## Geo redundancy failover simulation

## IIS Remote Management

# **Lesson learned and challenges faced**

## Stateful application might lose user session after scaling out

## Resource requirement

For users without cloud experience, it will difficult to really estimate the resources needed to be provisioned. One way is to conduct load test. However, the question could be how much could the real situation deviate from the load test, plus how much will it be considering the need for multiple load test. In short, where should we stop?

## Costing

Although cloud hosting seems reduce the Capex, it might end up even more costly than on-premises infra cost if resource planning and provisioning is not done properly.