Azure App Service - Logic Apps



Vision Scope

Prepared for

Microsoft

27-Jan-16

Version 1.0 Draft

Prepared by

**Ryan Irujo**

Senior Technical Consultant

ryan.irujo@lumagate.com

Contributors

**Add Contributors to Doc Properties**

Table of Contents

[1 Before you Begin 3](#_Toc441567010)

[2 Deploy an Azure Website with a SQL Database 4](#_Toc441567011)

[3 Walkthrough of Web App Features in Azure 12](#_Toc441567012)

[3.1 Configuring Diagnostic and Streaming Logs 12](#_Toc441567013)

[3.2 Configure Authentication / Authorization to your Web App 17](#_Toc441567014)

[3.3 Configure Backups 25](#_Toc441567015)

[4 Azure App Service Support (Preview) 26](#_Toc441567016)

[5 Continous Deployment using GitHub 31](#_Toc441567017)

[6 Additional Information 37](#_Toc441567018)

[6.1 Error: Unable to determine which solution file to build 37](#_Toc441567019)

1. Before you Begin

The objective of this Guide is to expose the participants to how they can implement Azure Service Apps in their existing Azure Subscription deployment model. At the time of this writing the current requirements to get started are below.

* Admin access to an Azure Subscription (minimum Trial Subscription)
* Visual Studio 2015 Installed (minimum Community Edition)
* Microsoft Azure PowerShell 1.0.1 (November 2015)
* Azure SDK 2.8.1 or higher

Finally, all Scripts, Templates and Applications mentioned within this Guide can be found in Lumagate’s GitHub Repository. If you require access, please contact a member of the Elite Incubation Team at Lumagate.

1. Creating a Logic App using the Azure Portal

In this section, you will be creating a simple Logic App in the Azure Portal that will list out the contents of a particular directory in a Dropbox account; as such, an existing Dropbox account is required to complete this module.

Login to the Azure Portal (<https://portal.azure.com>) and click on **New 🡪 Web + Mobile 🡪 Logic App (Preview)**.

Next, in the **Name** field, type in **mymaildroplogicapp**.

Next, under **App Service Plan**, click on **Or Create New** and type in **mymaildroplogicappplan**.

Next, make sure **Pricing Tier** is set to **S1**.

Next, click on **Resource Group** and then click on **Create a new resource group**, name the resource group **mymaildroplogicapp**.

Next, make sure the **Subscription** section is set to the Azure Subscription you wish to deploy to.

Next, change the **Location** section to the location you wish to deploy to.

Next, click on **Triggers and actions** and then click on the **Create from scratch** option.

Next, wait for the APIs to load from the right-hand side panel. Once they have loaded, click on the Office365 Connector, this will then populate into the main window and state, Connecting to the API… This process could take a few minutes to complete.

1. Additional Information

In this section, you will find related information to this Technical Guide that may be of use outside of the presented material.

If the customer does not have a properly configured VPN or Azure Subscription in place, please have them consider the Azure IaaS Foundations IP at http://aka.ms/MCS\_EPG\_Azure\_Iaas-Foundation.

These are just examples; update the requirements adapting to the specific customer situation if necessary. Points to consider might include: tool usage (MAP versus customer provided), server locations, availability of test lab, and so on.