

Lab 4: Using Azure API Management

Hands-on Lab

Released:

Terms of Use

© 2018 Microsoft Corporation. All rights reserved.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see **Microsoft Copyright Permissions** at [**http://www.microsoft.com/permission**](http://www.microsoft.com/permission)

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The Microsoft company name and Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

**This document reflects current views and assumptions as of the date of development and is subject to change.  Actual and future results and trends may differ materially from any forward-looking statements.  Microsoft assumes no responsibility for errors or omissions in the materials.**

**THIS DOCUMENT IS FOR INFORMATIONAL AND TRAINING PURPOSES ONLY AND IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.**

Table of Contents

[Lab Overview 1](#_Toc508731806)

[Exercise 1: Create a Logic App and deploy it in API Management 2](#_Toc508731807)

# Lab Overview

###### Abstract

In this lab, we will be creating an instance of Azure API Management. We will be enabling a few security features of the product on a sample API.

###### Learning Objectives

After completing the exercises in this lab, you will be able to:

* Create a Logic App and deploy it in API Management.

**Estimated time to complete this lab: *20* minutes**

# Exercise 1: Create a Logic App and deploy it in API Management

#### Scenario

In this exercise, you will be creating a Logic App with an HTTP trigger so that this can be used as an API for our API Management lab.

After completing this exercise, you will understand:

* How to create a Logic App in the portal designer
* How to host an API with API Management

#### Prerequisites:

To complete this lab, you should have a basic understanding of the Azure Portal.

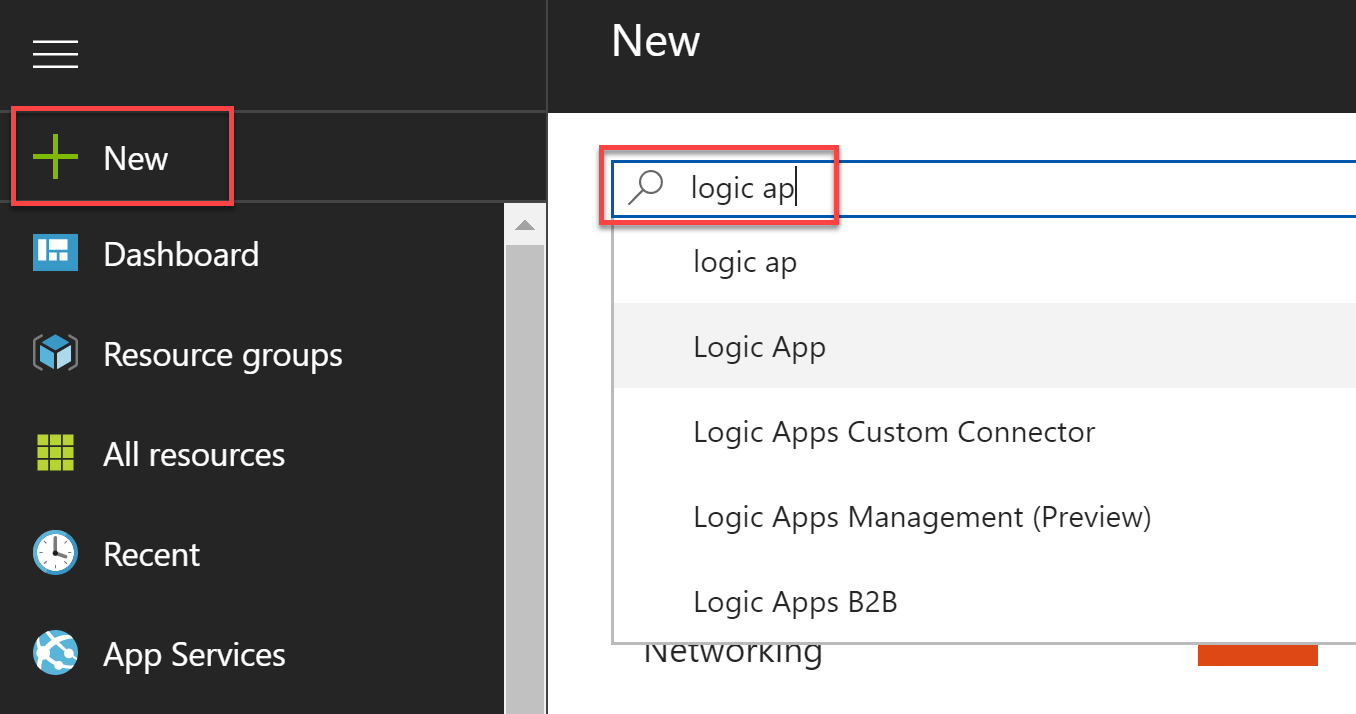
#### Install and configure:

You will need:

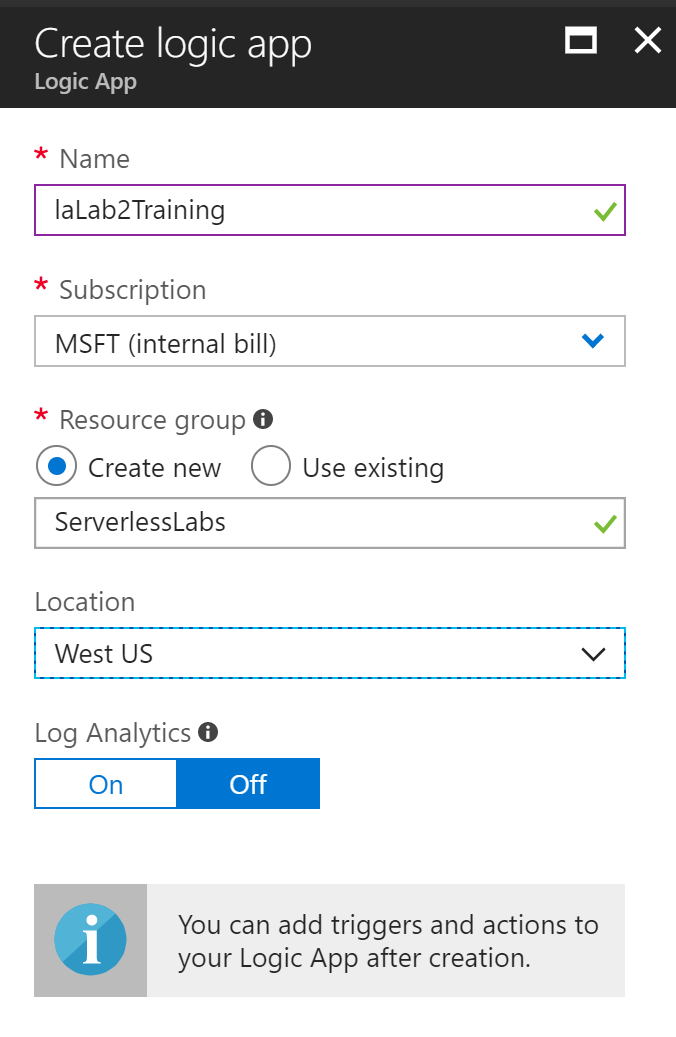
* A desktop computer (host) running Windows 10
* An active Azure subscription (MSDN credits will suffice)
* An active Internet connection
* Postman

#### Creating a new Logic App

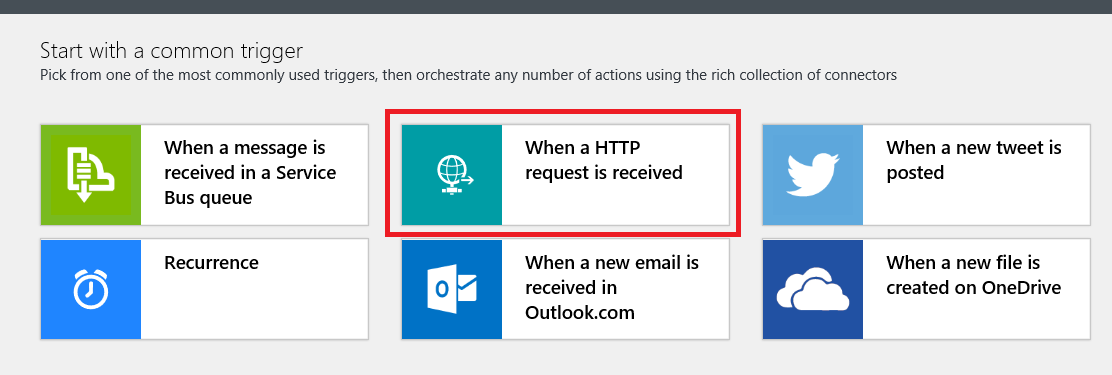
1. Navigate to the Azure Management portal, <http://portal.azure.com>.
2. Select **+ New**, then type **logic app** into the search box on top. Select **Logic App** from the results.



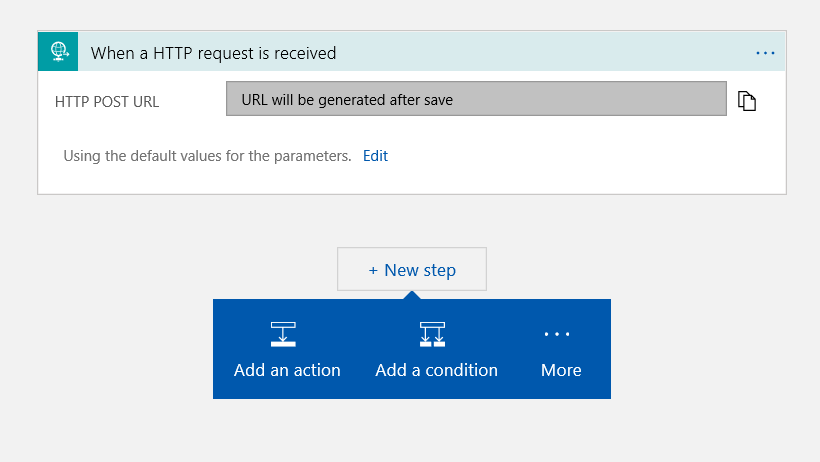
1. Select the **Create** button on the Logic App overview blade.
2. On the Create Logic App blade, specify the following configuration options:
   1. **Name**: unique value for the App name (ensure the green check mark appears).
   2. Specify the Resource Groupusing one that will keep your resources for the labs organized.
   3. Select the **Location** that is appropriate for your subscription.



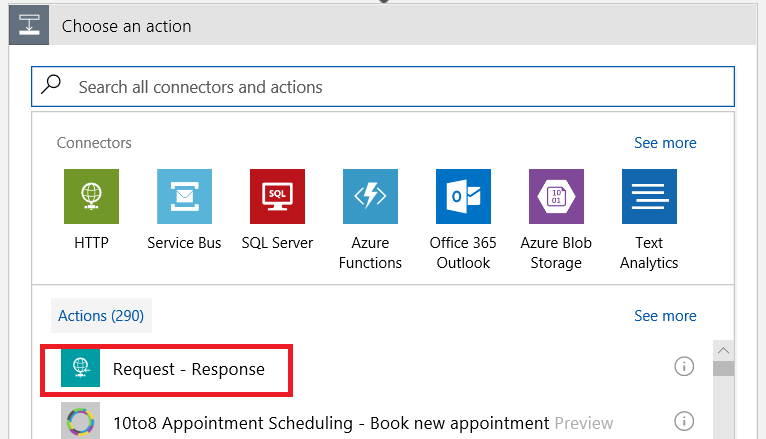
1. Open the Logic App once it has been provisioned, click the **Edit** button on the **Logic App Designer** link to open the designer.
2. Select the option “**When a HTTP request is received**” from the prebuilt templates.



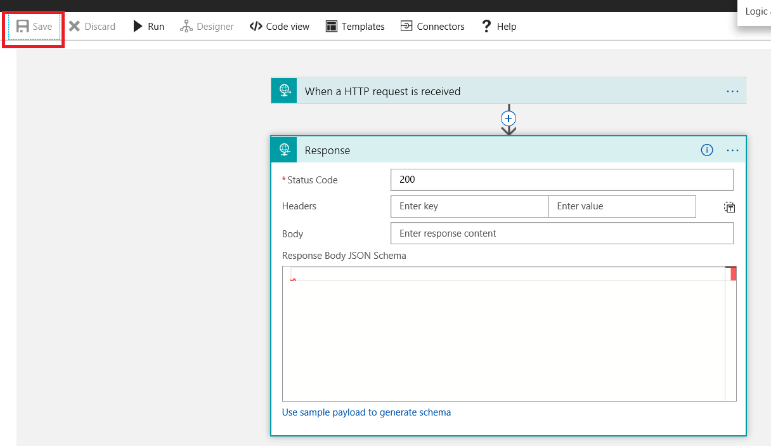
1. Click on Add an action



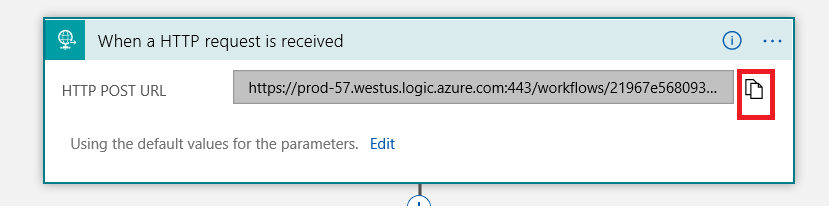
1. From a list of actions select “Request-Response”



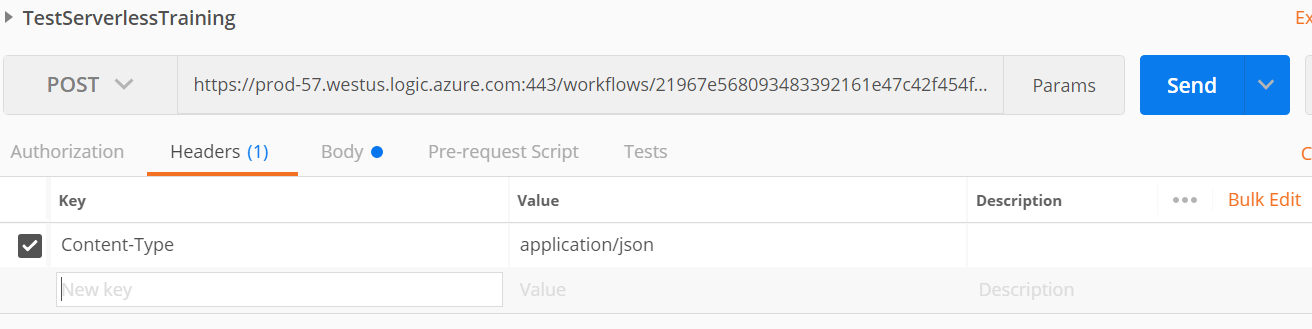
1. Click on Save



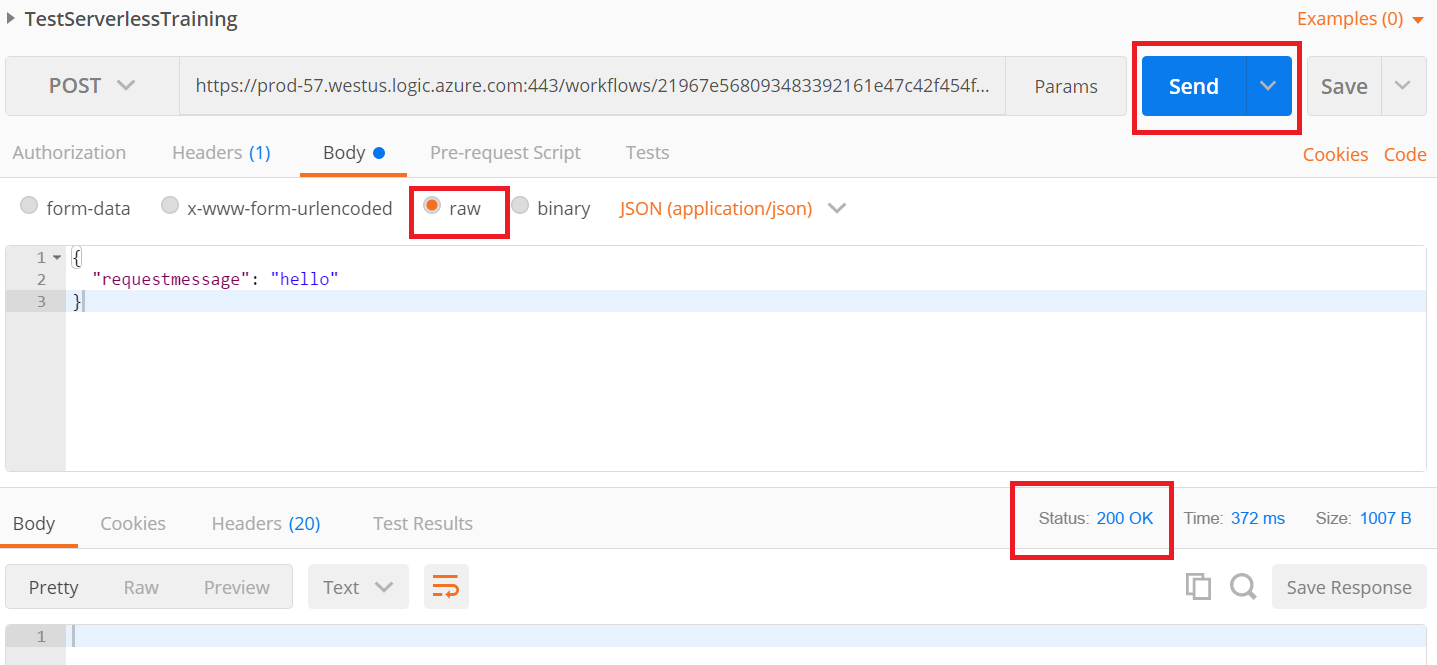
1. Copy the HTTP Post URL generated after saving the logic app and test the API using Postman



1. Start Postman and create a new request. Set it for POST and use the URL that was copy from the previous step.

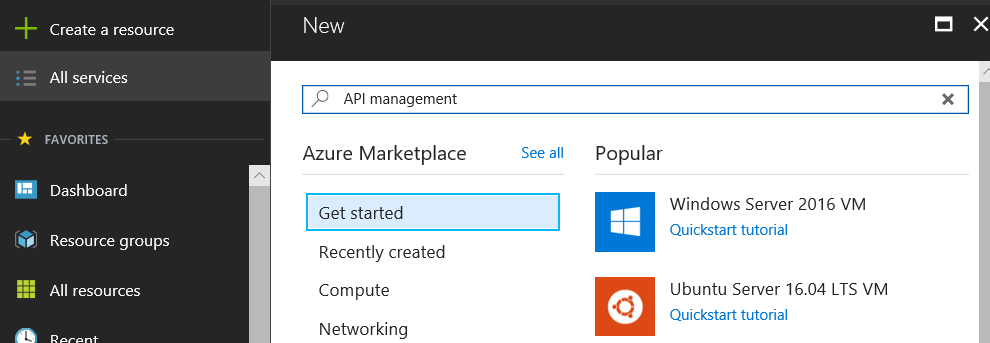


1. Note that in the Header section you should add the “Content-Type” key and add its value as “application/json”
2. You can see that you have successfully called the Logic App with Postman. A few things to note in the request that is sent using Postman :
   1. In the Body section the request message is of “raw format”
   2. The status returned by the logic app is “200”

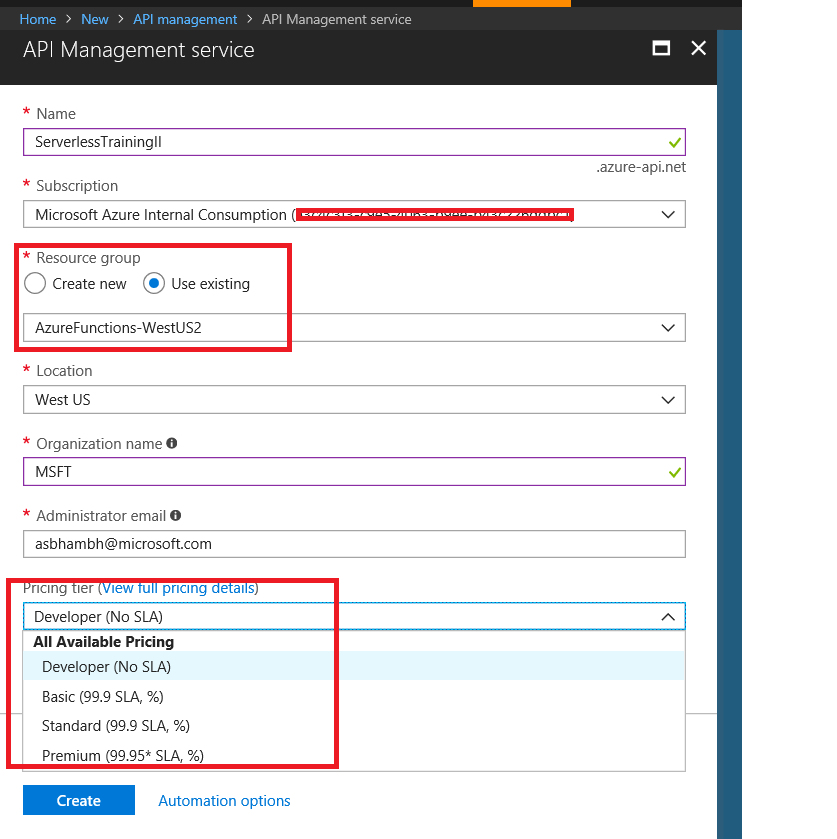


#### Create API Management instance

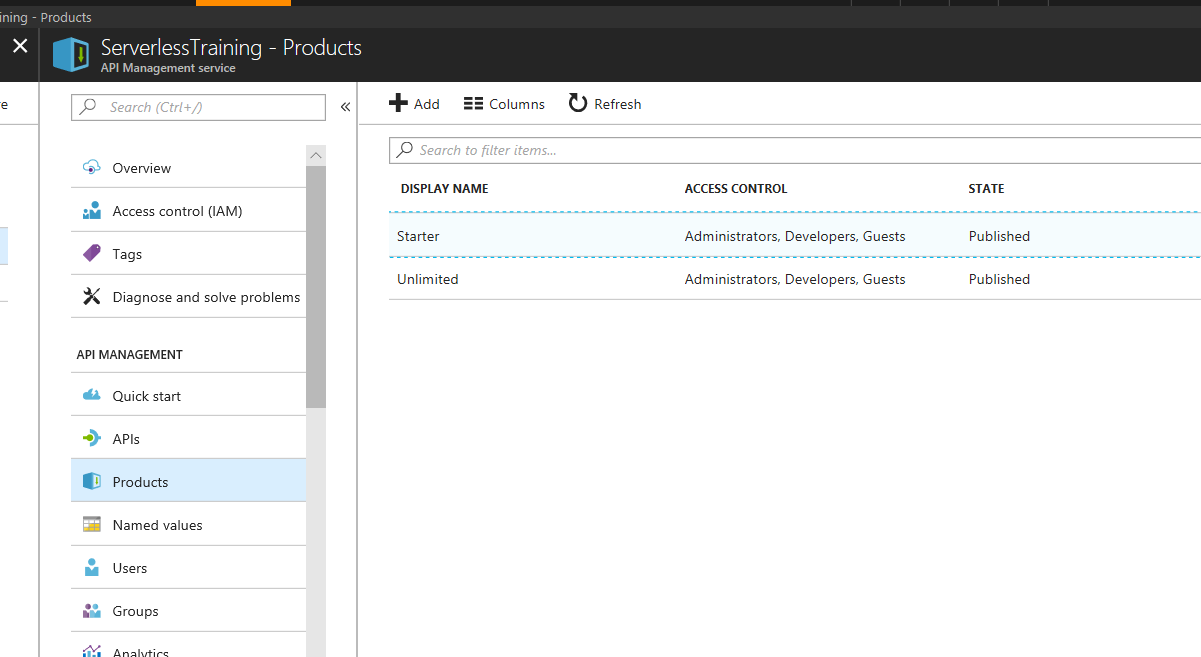
1. Next you will create an API Management instance, navigate to the Azure Management portal, <http://portal.azure.com>.
2. Select + New, then type API Management into the search box on top. Select API Management from the results.



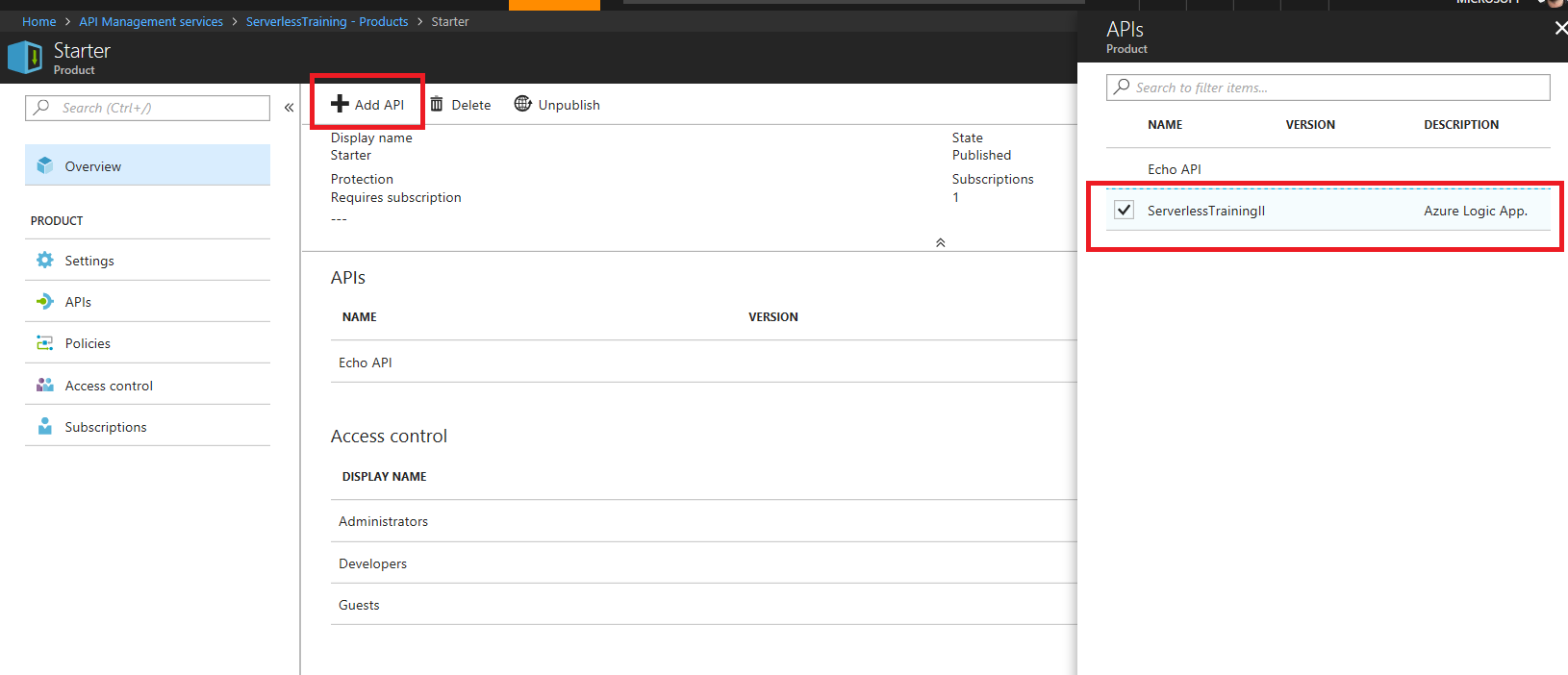
1. Provide the following details for your instance of API Management. You can use the same resource group that you have been using thus far.
2. Also note the various pricing options and select “Developer”



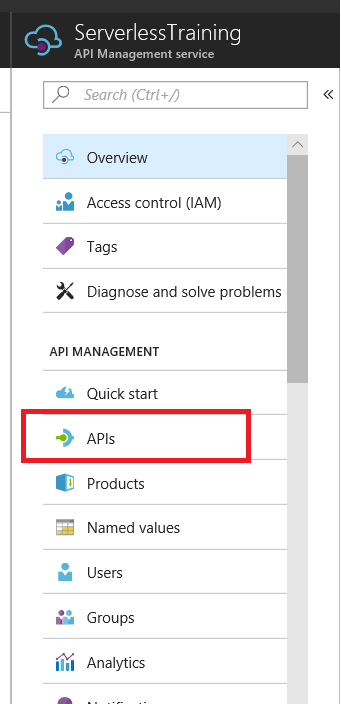
1. Select Products from the API Management blade and click on Starter on the right pane



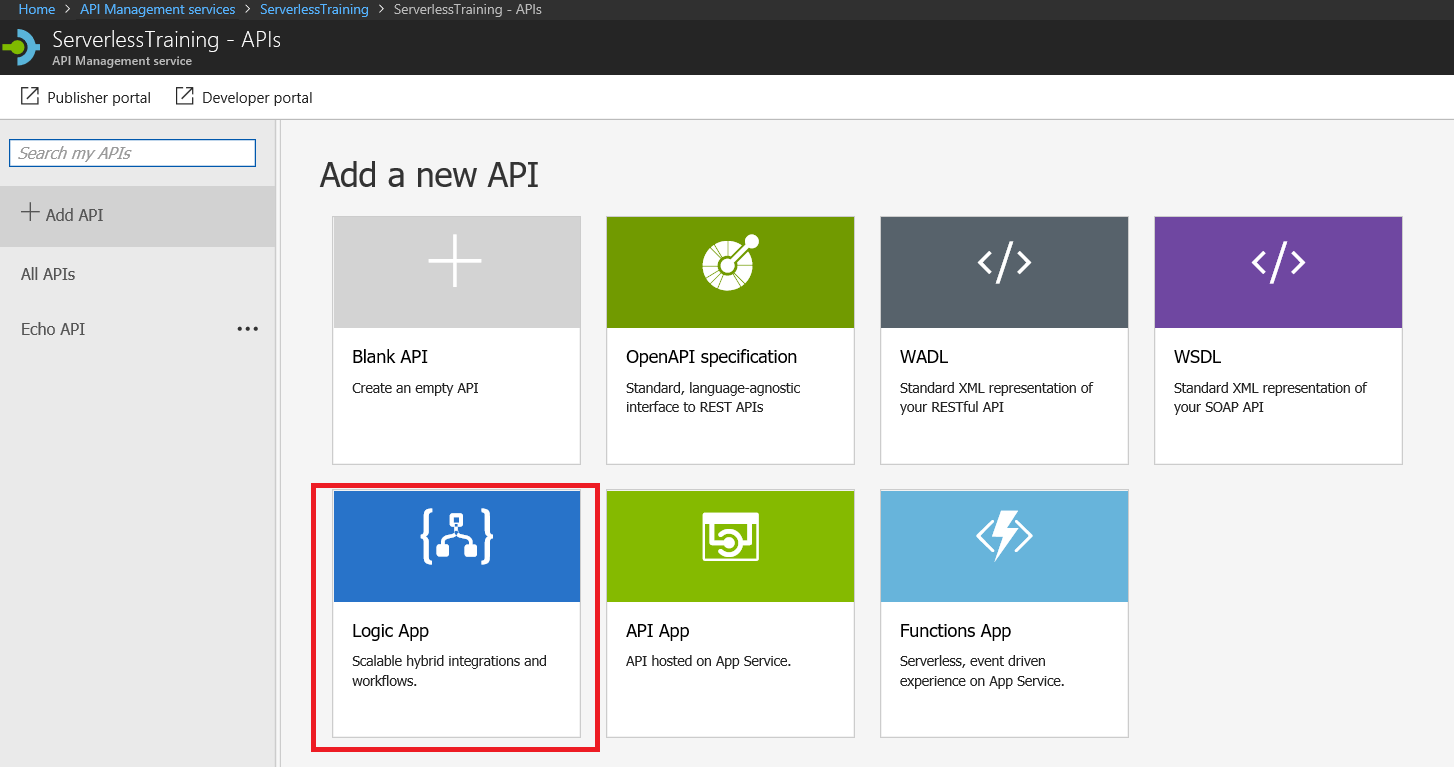
1. Click on Add API and select “ServerlessTraining” API



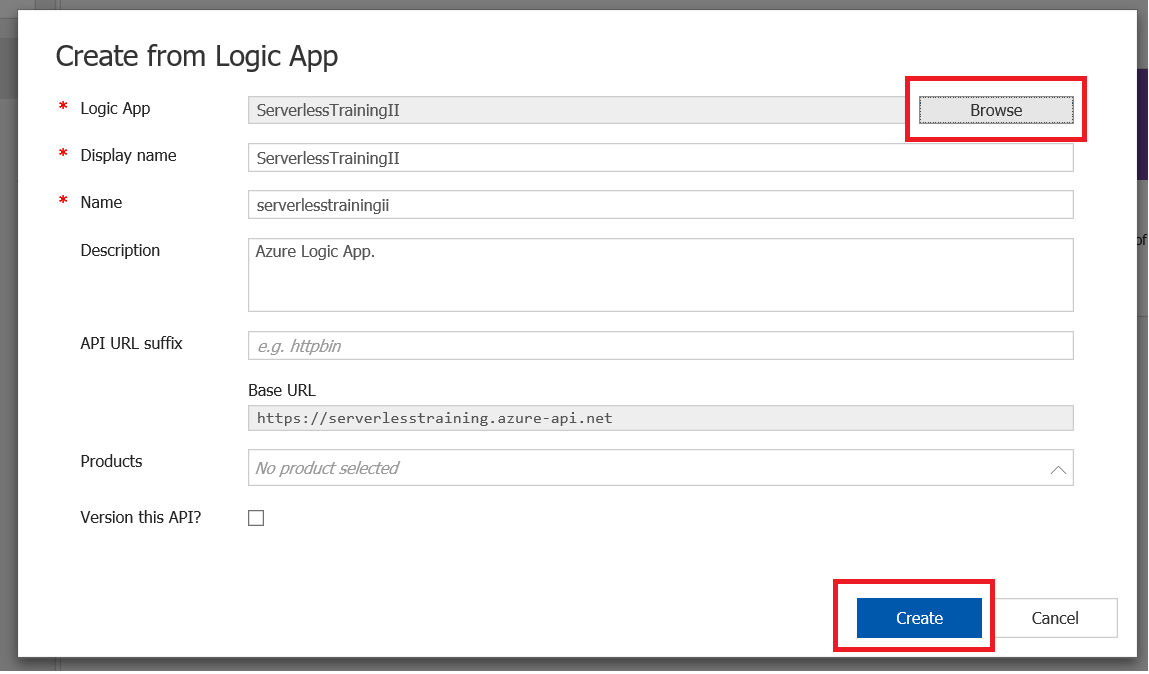
1. Do the same steps for the “Unlimited” product as well
2. Open your API Management instance’s blade and select APIs



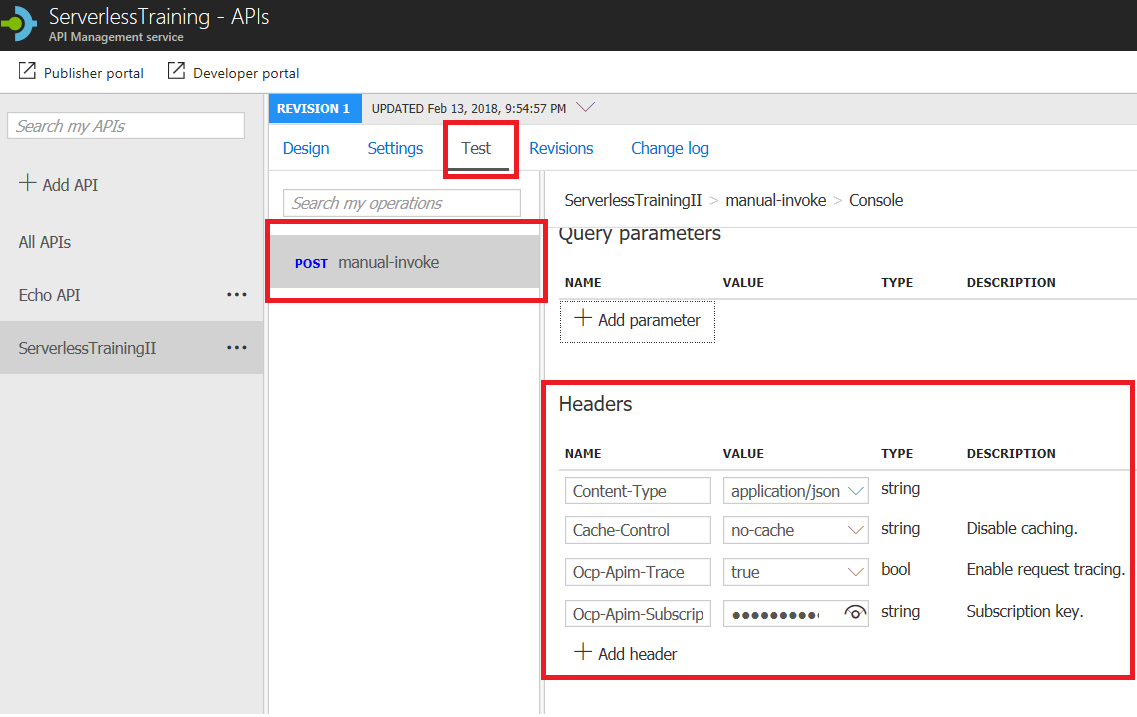
1. In this blade select Logic App



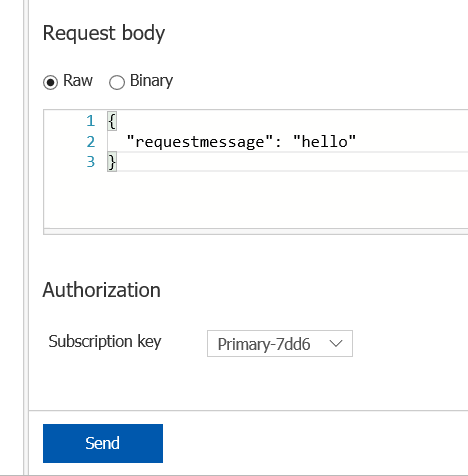
1. Select the Logic App you just created, and the rest of the details will be self-populated



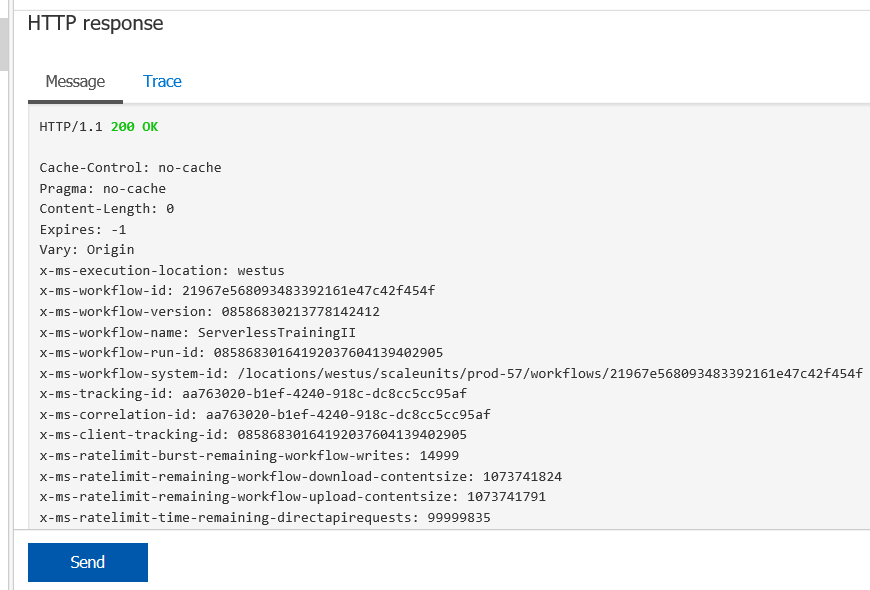
1. Click on create
2. After this navigate to the “Test” button and you should see this:



1. In the request body text box put the below value and click on “Send”



1. The message is sent successfully, and we receive a response status of “200”



1. You can see here that API Management was used to call the Logic App as well. On could then use API Management to manage the http interface for the Logic App.