Лабораторная #5.

Начало работы с WebJobs, Logic Apps и Service bus.

В этой работе мы познакомимся с основами разработки event-driven web-приложений в Azure. Мы разберём как деплоить WebJobs. На портале создадим Logic App. Создадим и будем использовать Service Bus для связи компонентов.

Что конкретно мы сделаем:

* Создадим ресурс *Service bus*
* Развернём наш собственный *Web Job* в облаке, который будет читать сообщения из *Service bus*
* Создадим *Logic App,* который будет тригериться от запросов из *Azure Web Job* и отправлять юзерам электронные
* Проверим, что мы действительно сделали всё верно добавив пару картинок в наше приложение. Результат – Email на нашу почту

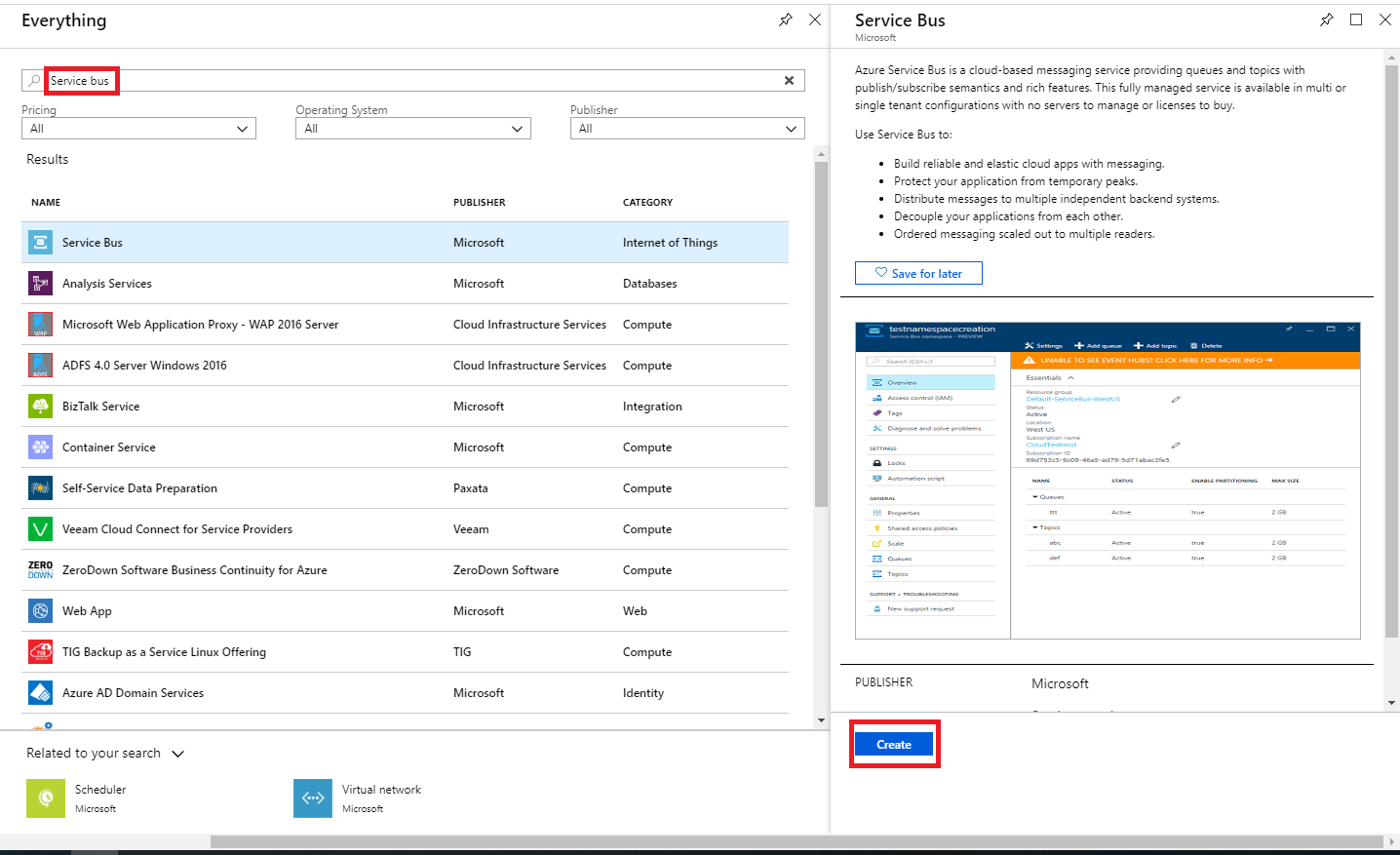
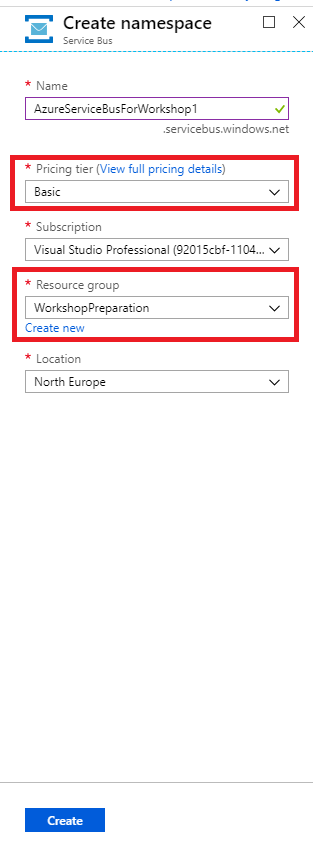
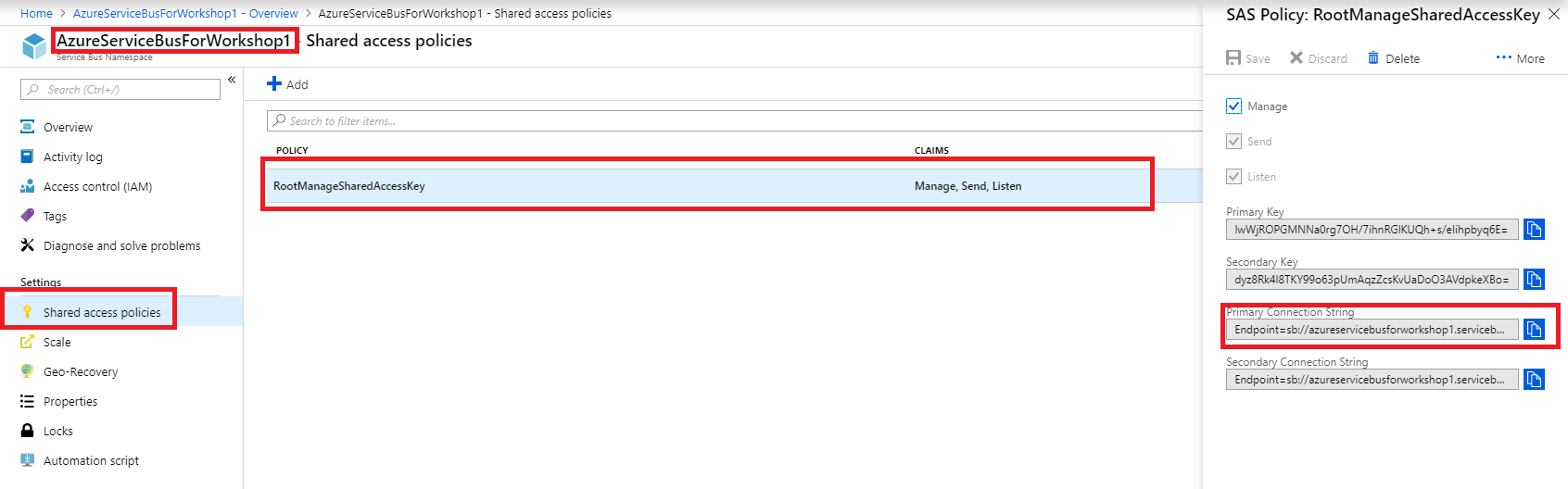
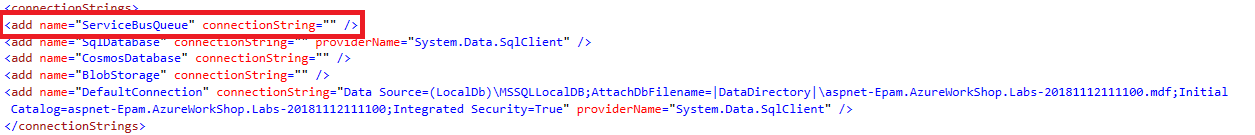
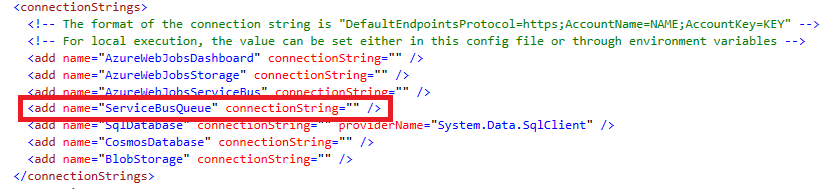
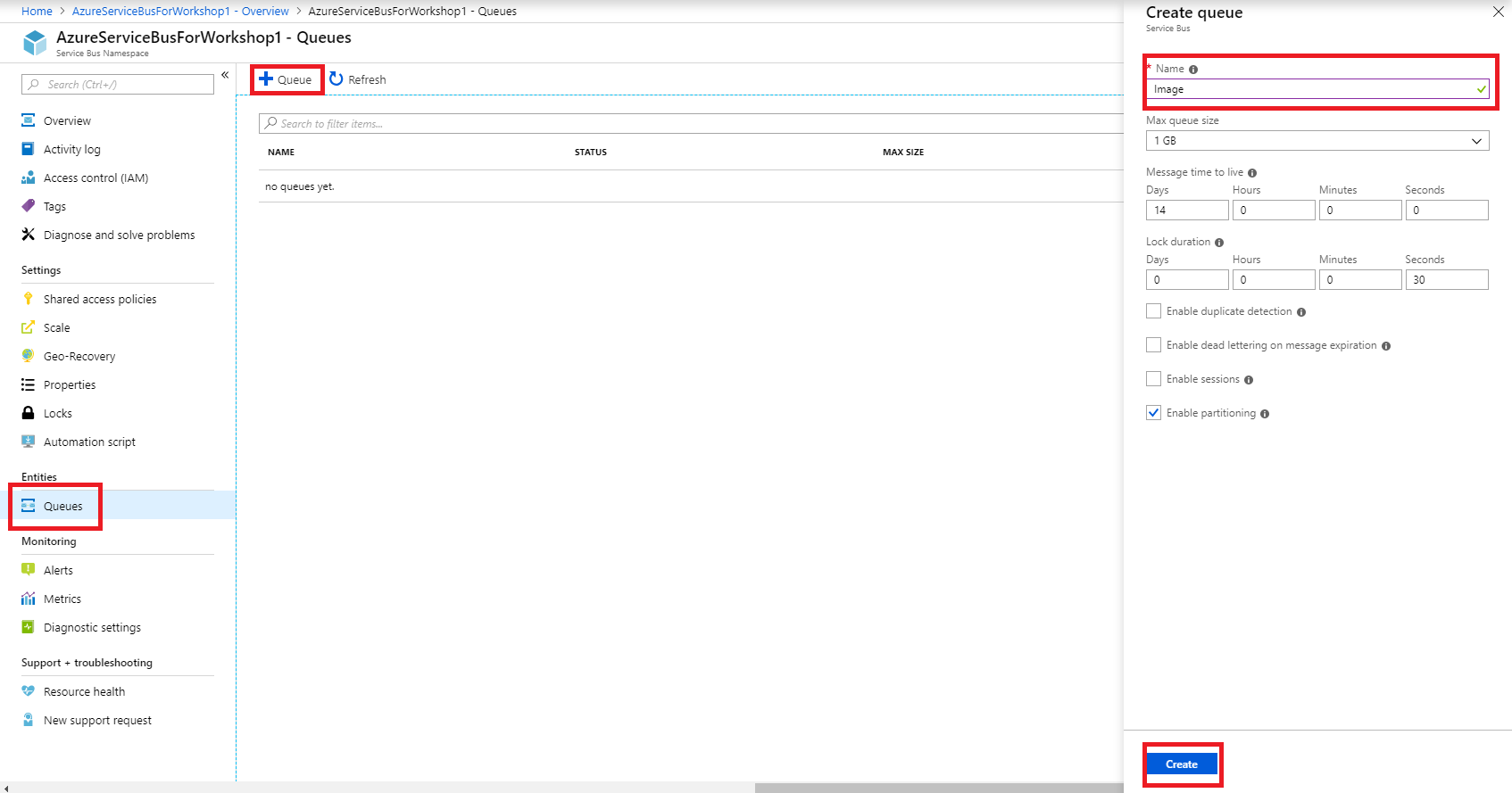
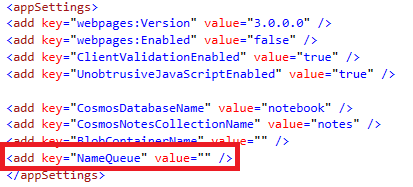
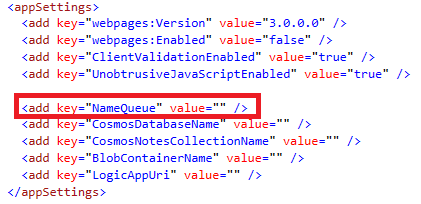
Примечание!

В данной работе, а также в ряде последующих, нам потребуется использовать уникальные имена для сервисов в Azure (для Web App, Storage Account, …). В некоторых случаях инструменты предлагают сгенерировать эти имена автоматически, но увы – не всегда. Поэтому, заранее придумайте такое уникальное имя, например по схеме:

AzureWorkshop<ВашеИмя><ДатаРождения/ИлиПростоНаборЦифр>

# Задание 01. Создание ресурса Service Bus

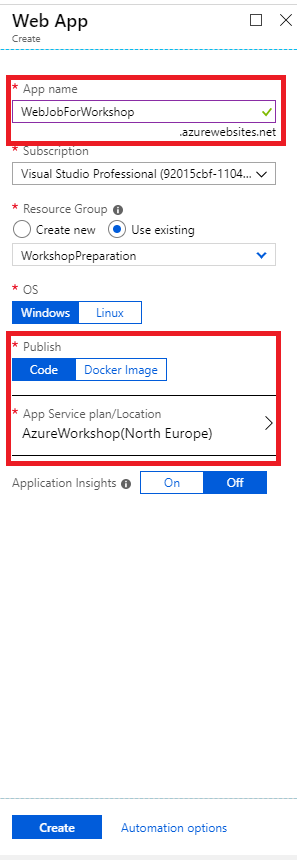
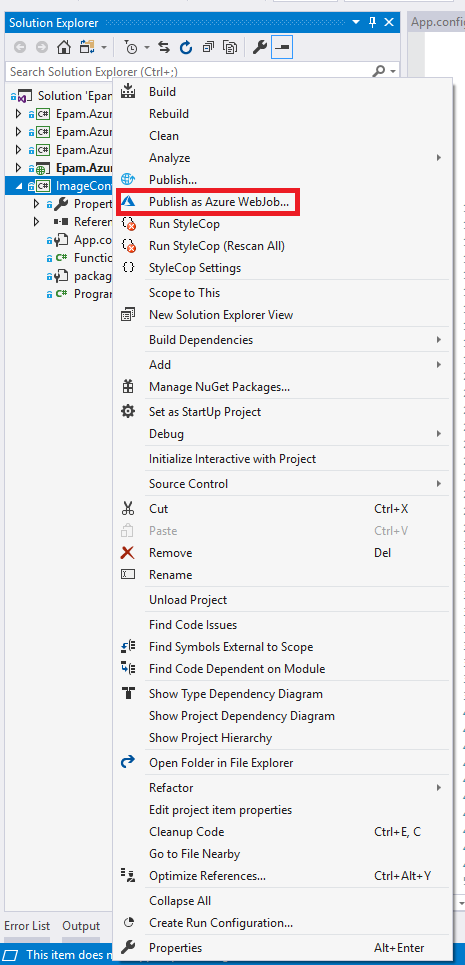
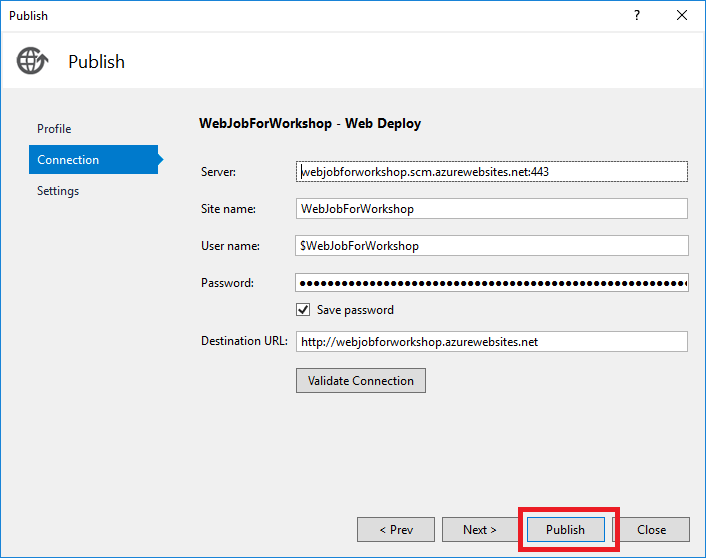
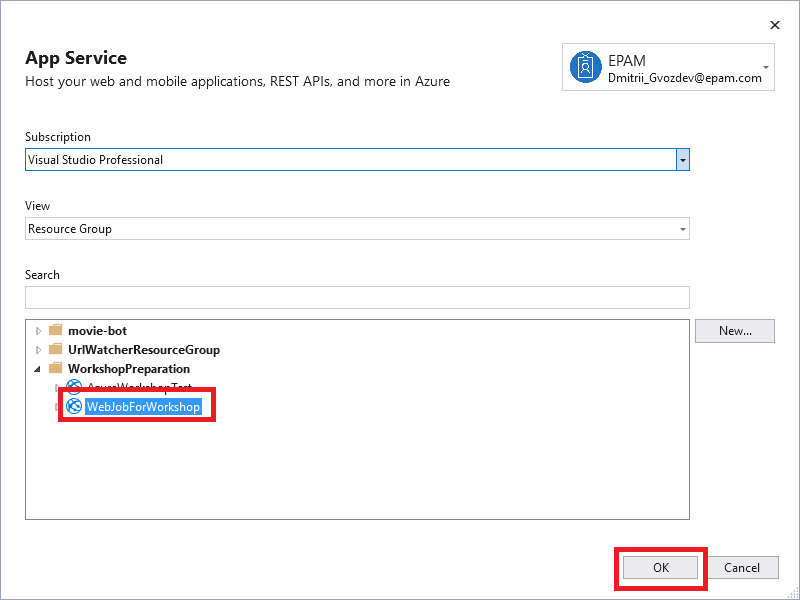
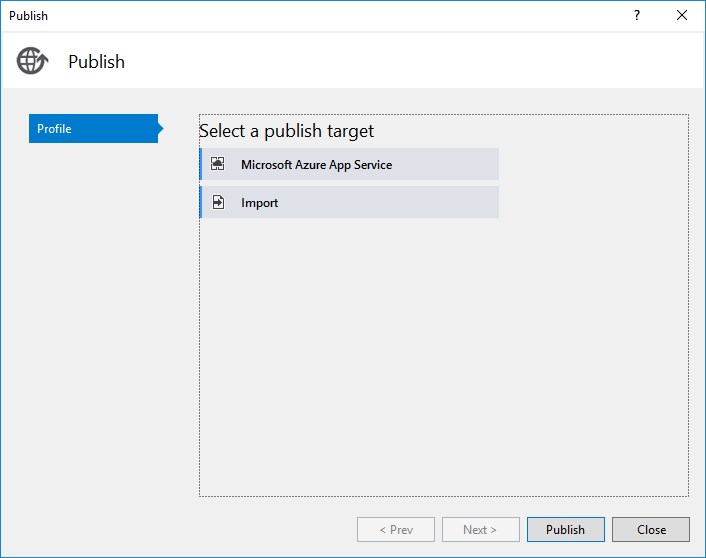
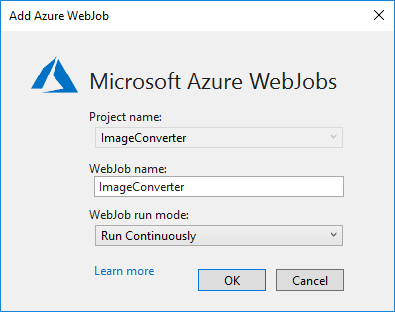
В этом задании мы создадим ресурс Service Bus на портале Azure. И подключимся к нему из нашего приложения (оно будет добавлять messages в очередь) и из новой Web Job (она будет читать messages из очереди).

* Создадим Service Bus по уже знакомому нам алгоритму:  
    
    
  
* После этого для того, чтобы подключить к этому Service Bus нам нужно получить ConnectionString для этого ресурса. Для этого нужно открыть страницу с этим Service Bus; В левых вкладках выбрать пункт Shared access policies, потом выбрать policy RootManageSharedAccessKey. Из неё можно будет достать главную ConnectionString.  
  
* Скопируем эту строку в Web.config нашего приложения  
  
* Скопируем эту строку в App.config новой WebJob  
  
* Теперь нам хочется начать передавать сообщения через очередь внутри Service Bus. Для начала эту очередь нужно создать. Заметьте, что здесь нам очень важно сохранить имя очереди. Мы будем использовать далее.  
  
* Добавим это название в Web.config  
  
* Добавим это название в App.Config  
  

Таким образом мы подключили Service Bus!  
Можно попробовать запустить приложение и добавлять записки. Пока что приложение будет падать по ошибке, но в Service Bus будут появляться новые messages.

# Задание 02. Деплоймент новой WebJob

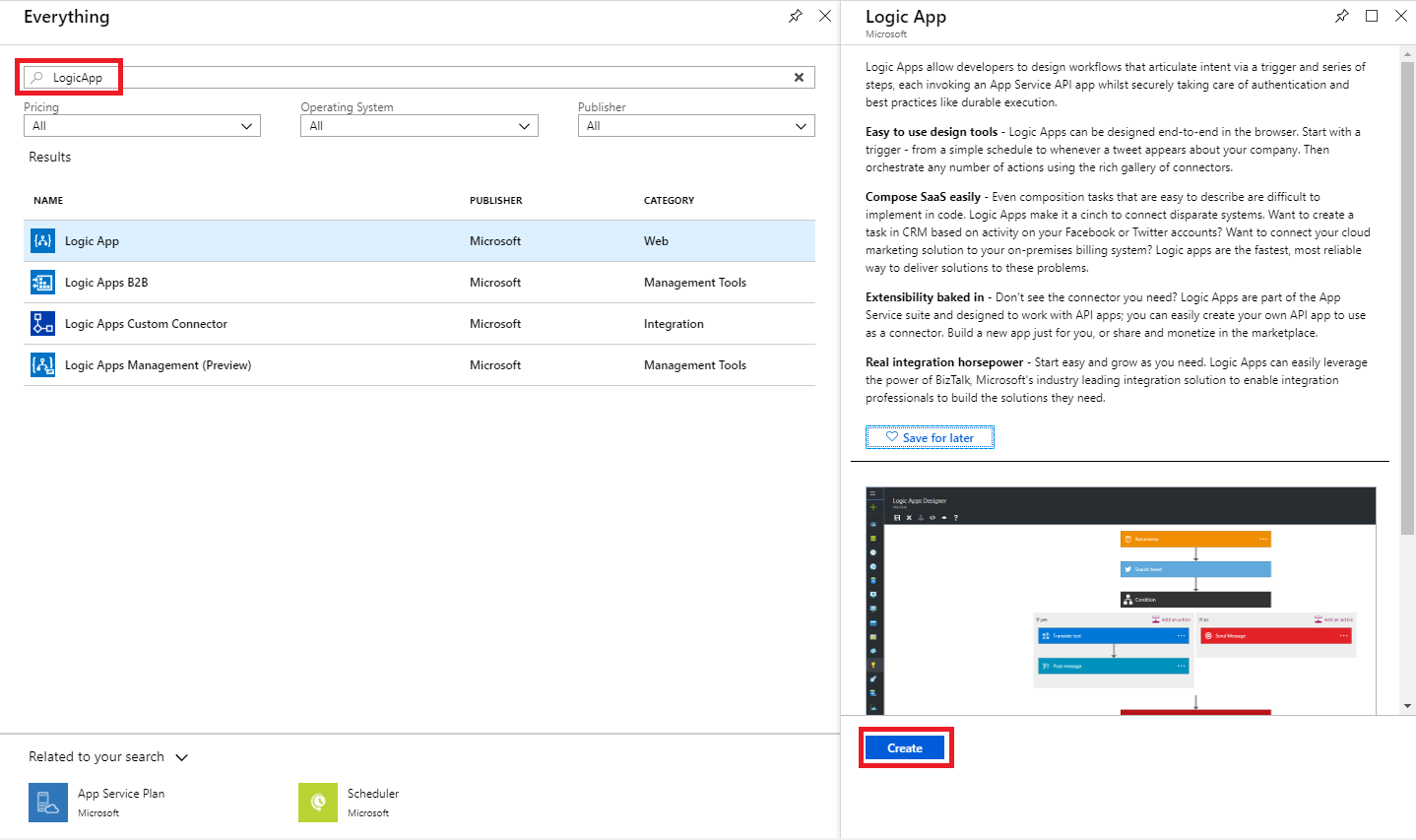
В этом задании мы создадим WebJob на портале. И при помощи Visual Studio развернём нашу новую WebJob в Azure

* Для начала мы создадим Web App по уже хорошо известному нам алгоритму:  
  
* Вернёмся в Visual Studio. Кликнем на новую WebJob *(ImageConverter)* правой кнопкой мыши. В выпавшем диалоговом окне выберем Publish as Azure WebJob.  
  
* Пройдём по следующим диалоговым окнам как показано на следующих скриншотах:  
  

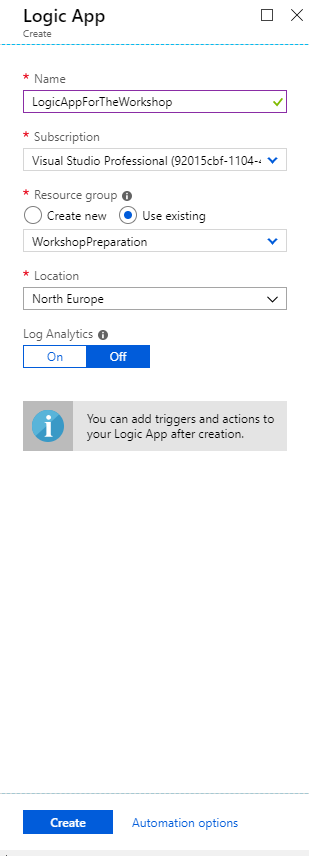
# Задание 03. Создание Azure Logic App

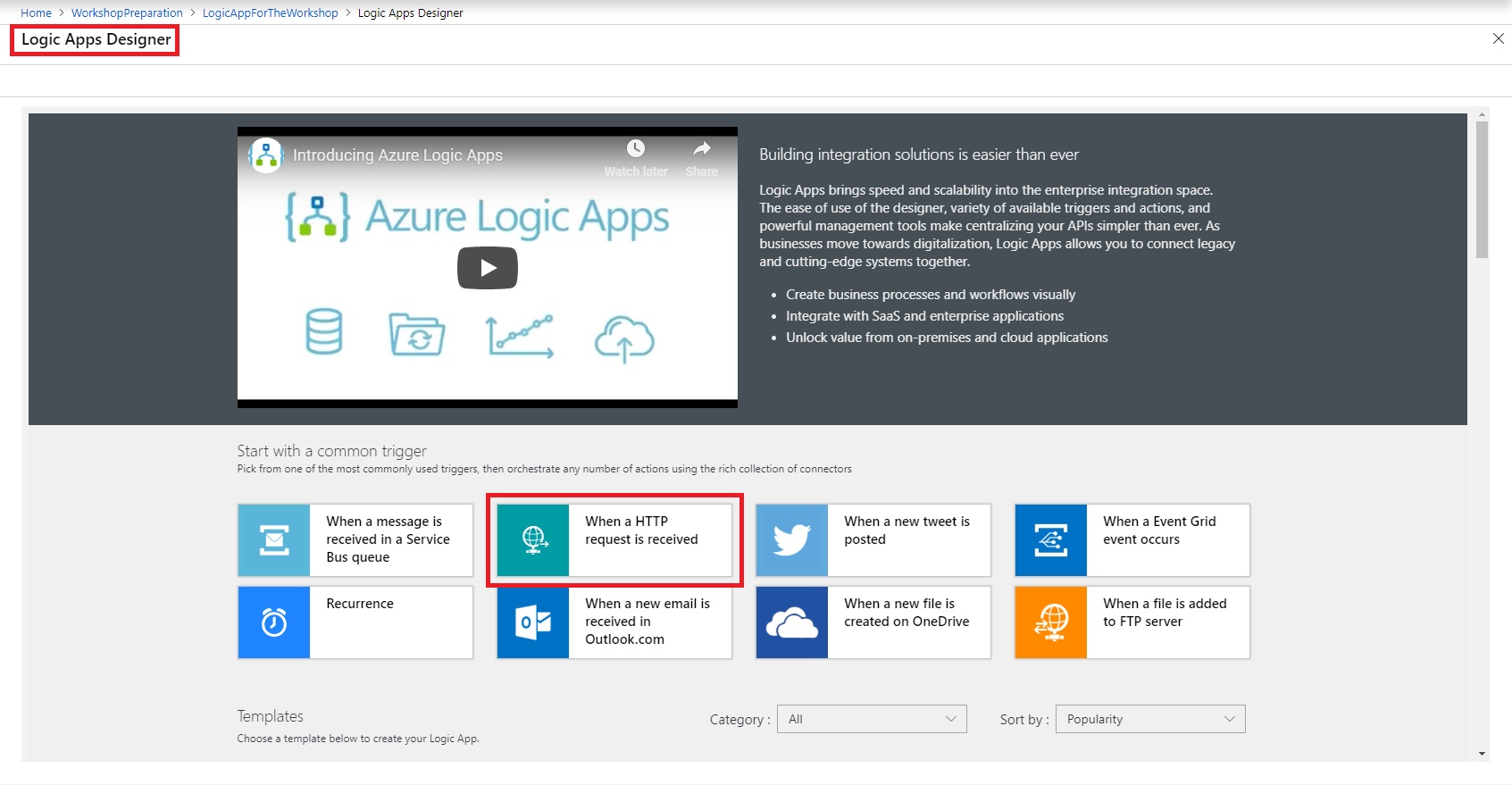
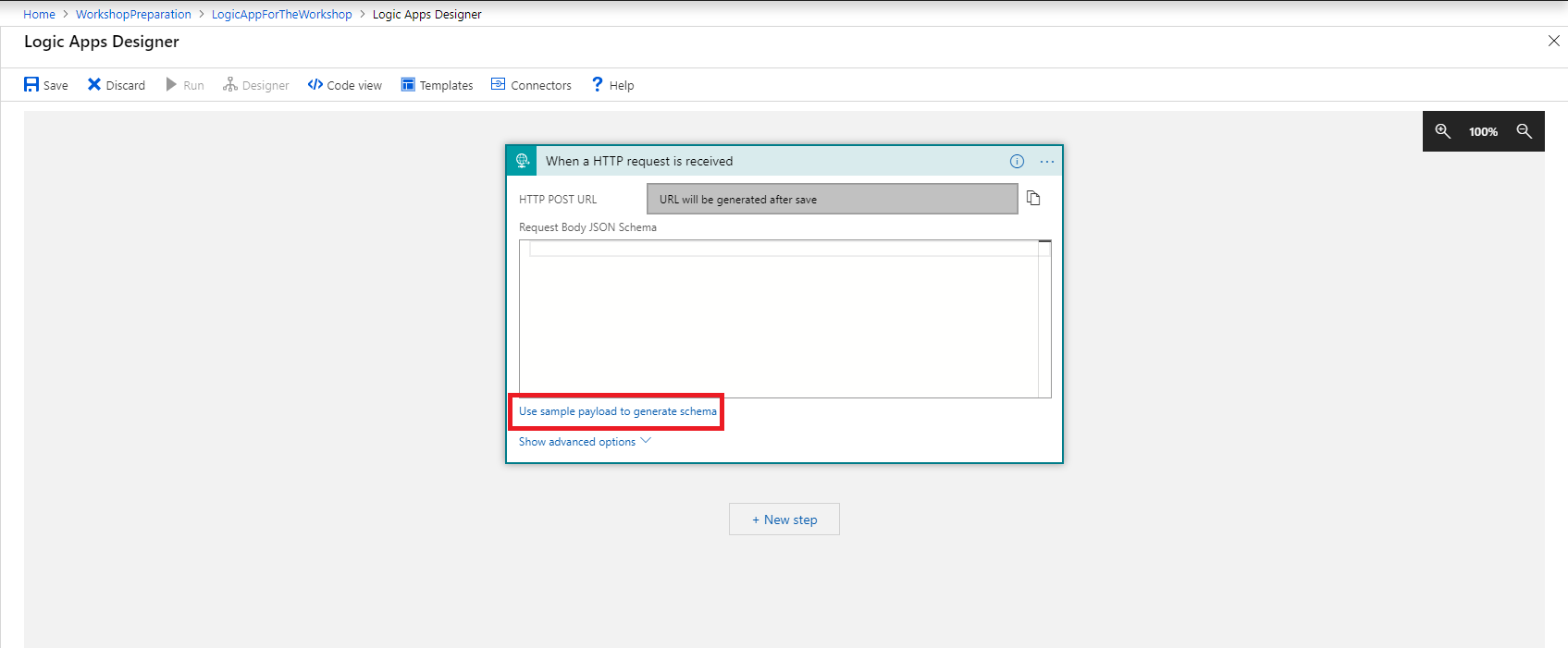
В этом задании мы создадим Azure Logic App на портале.

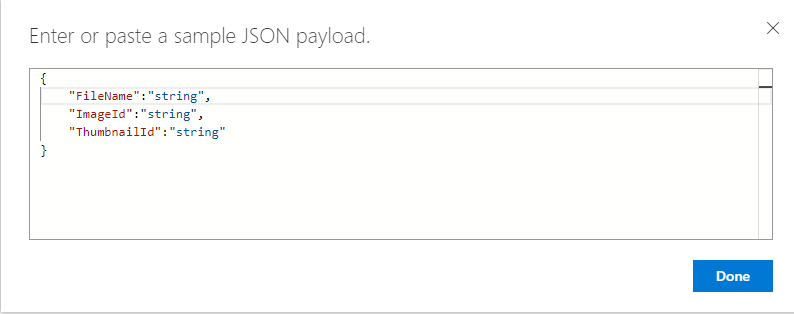
* Перейдем в *Create a resource* в поиске введем и нажмем create.



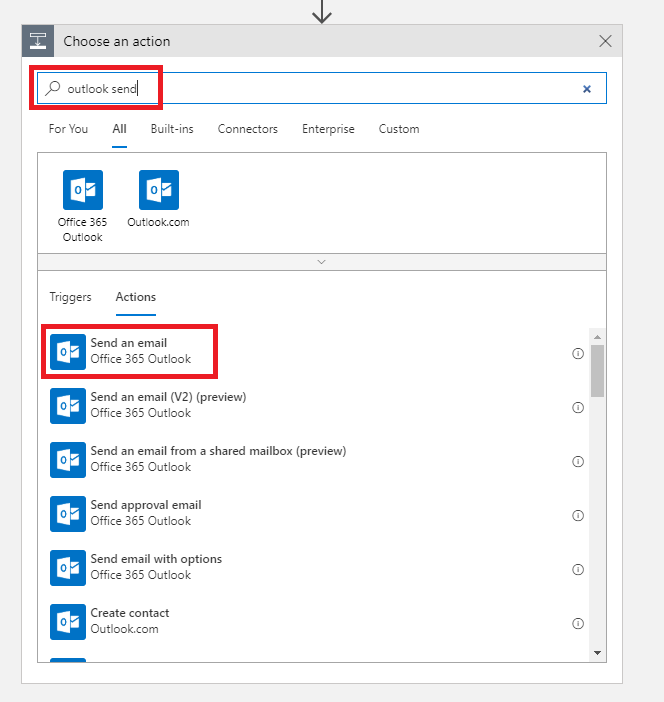
* Далее перейдем в меню создания и заполним обязательные поля.



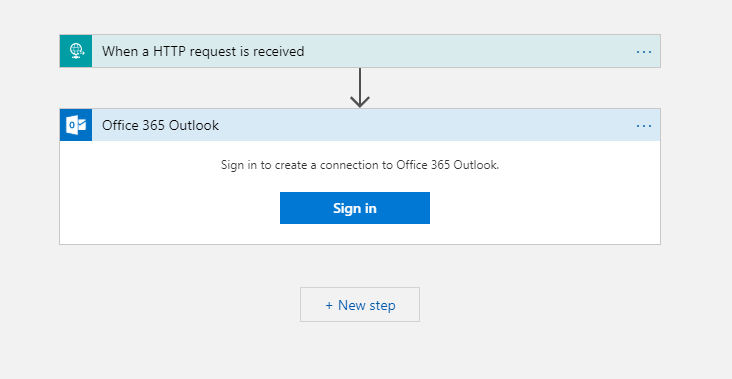
* **После создания перейдем в *Logic apps Designer* и выберем *When a HTTP*
* Далее откроется форма и нам нужно будет указать схему.
* Схема сгенирируется атоматически, если мы опишем параметры запроса.



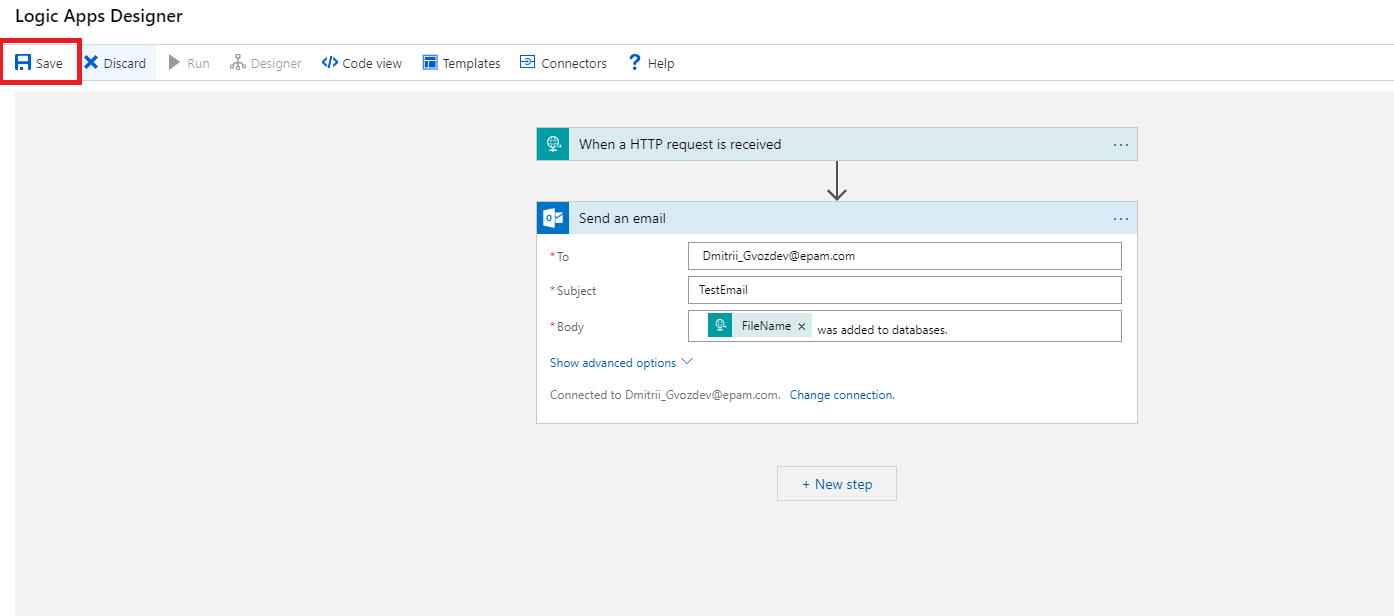
* Далее наждем добавить новый степ. И введем в поиск outlook send. *Выберем send email.*



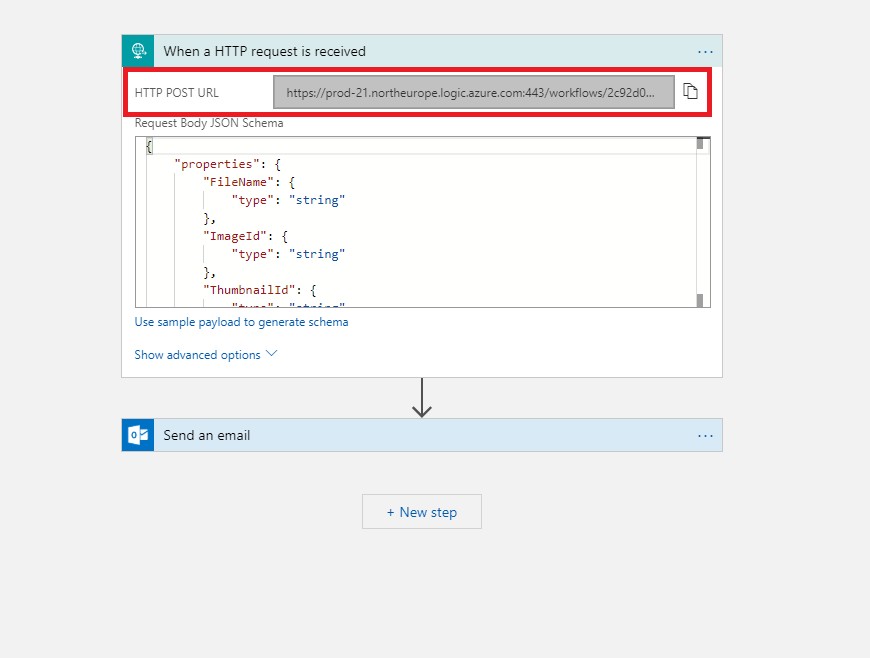
* Необходимо войти в учетную запись отправителя писем.



* Введем кому будут слаться уведомления и тему письма и выберем параметр FileName из параметро Json.



* Скопируем URL запроса.



* Далее в файл web.confic добавим ссылку на наше Logic App.

