AZ-203 Developer Training

Connect to and Consume Azure Services

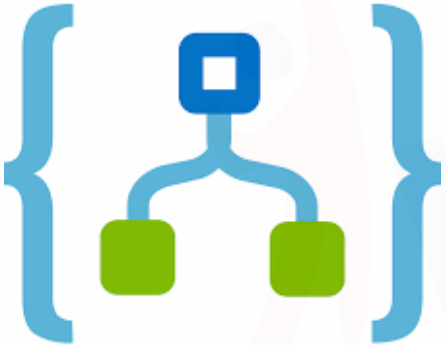
Agenda

- What is Azure Logic Apps?
- How does Logic Apps work?
- Why use Logic Apps?
- Create Logic Apps - Apps Portal
- Create Logic Apps - Visual Studio
- Connectors
- Custom Connectors
- Azure Search
- Azure API Management
- Azure Event Grid
- Azure Notification Hub

What is Azure Logic Apps?

What is Azure Logic Apps?

Azure Logic Apps



Azure Logic Apps is a framework that simplifies designing complex workflows

- It is a cloud service that helps you schedule, automate, and orchestrate tasks, business processes, and workflows when you need to integrate apps, data, systems, and services across enterprises or organizations.
- Logic Apps simplifies how you design and build scalable solutions for app integration, data integration, system integration, enterprise application integration (EAI), and business-to-business (B2B) communication, whether in the cloud, on premises, or both.

How does Logic Apps work?

How does Logic Apps work?

Every logic app workflow starts with a trigger, which fires when a specific event happens, or when new available data meets specific criteria.

Many triggers provided by the connectors in Logic Apps include basic scheduling capabilities so that you can set up how regularly your workloads run.

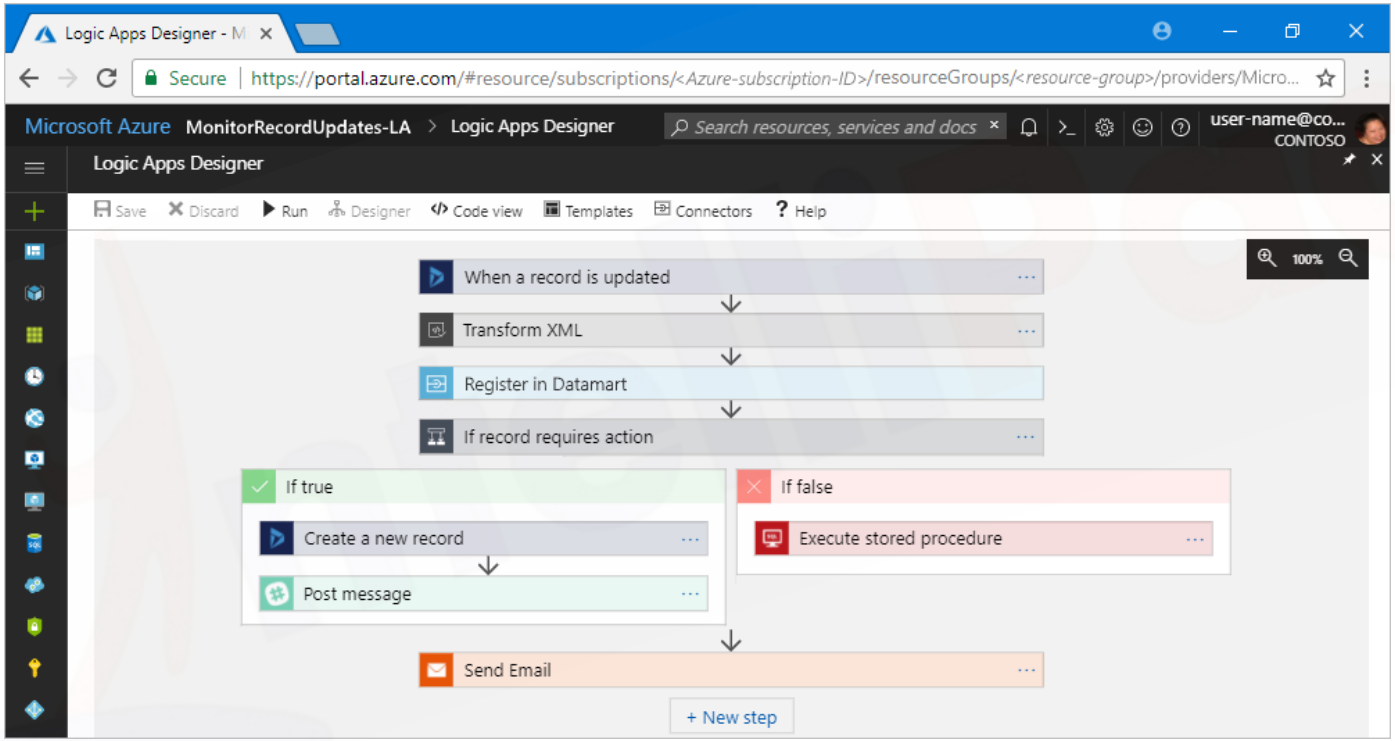
Each time that the trigger fires, the Logic Apps engine creates a logic app instance that runs the actions in the workflow.

These actions can also include data conversions and flow controls, such as conditional statements, switch statements, loops, and branching.

You can build your logic apps visually with the Logic Apps Designer, which is available in the Azure portal through your browser and in Visual Studio.

For more custom logic apps, you can create or edit logic app definitions in JavaScript Object Notation (JSON) by working in the "code view" editor.

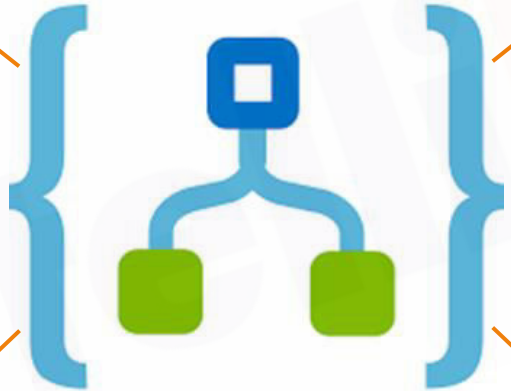
How does Logic Apps work?



Why use Logic Apps?

Why use Logic Apps?

Here are some reasons why organizations should use Azure Logic Apps:



- Visually build workflows with easy-to-use tools
- Get started faster with logic app templates
- Write once, reuse often
- Pay only for what you use
- Built-in extensibility
- No upfront setup cost

Why use Logic Apps?

Here are some reasons why organizations should use Azure Logic Apps:

Visually build workflows with easy-to-use tools:

With the use of visuals, the code is reduced and the logic can be reused. New composite workflows can either be created or modified without having to wait for application vendors.

Get started faster with logic app templates:

Templates range from simple connectivity for software-as-a-service (SaaS) apps to advanced B2B solutions plus "just for fun" templates.

Write once, reuse often:

Create your logic apps as templates so that you can deploy and reconfigure your apps across multiple environments and regions.

Why use Logic Apps?

Here are some reasons why organizations should use Azure Logic Apps:

Pay only for what you use:

Logic Apps uses consumption-based pricing and metering unless you have logic apps previously created with App Service plans.

Built-in extensibility:

If you don't find the connector that you want or need to run custom code, you can extend logic apps by creating and calling your own code snippets on-demand through Azure Functions.

No upfront setup cost:

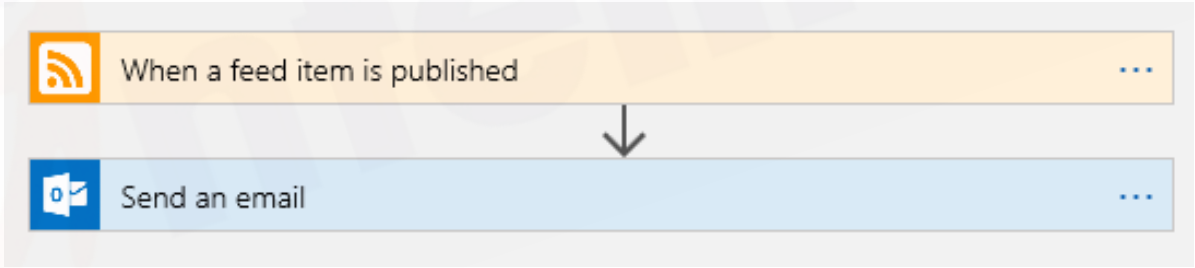
With Logic Apps as a managed Microsoft Azure Cloud service, there is no upfront setup or infrastructure cost. Organizations pay only when a particular app is executed.

Create Logic Apps - Apps Portal

Create Logic Apps - Apps Portal

This quickstart introduces how to build your first automated workflow with Azure Logic Apps. Here you create a logic app that regularly checks a website's RSS feed for new items.

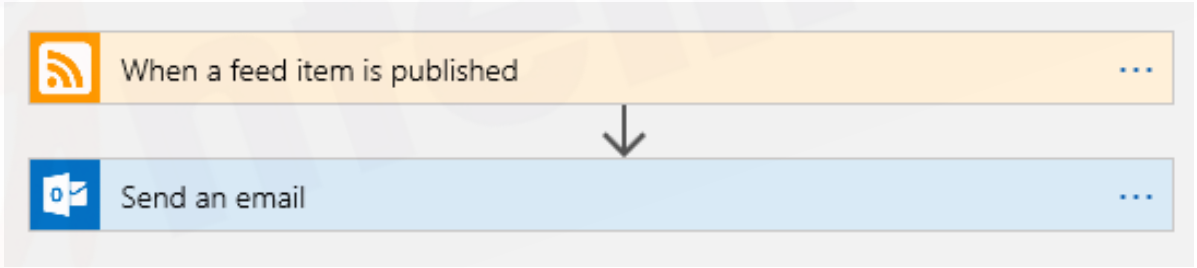
If new items exist, the logic app sends an email for each item. When you're done, your logic app looks like this workflow at a high level:



Create Logic Apps - Apps Portal

To follow this quickstart, you need an email account from a provider that's supported by Logic Apps, such as Office 365 Outlook, Outlook.com, or Gmail.

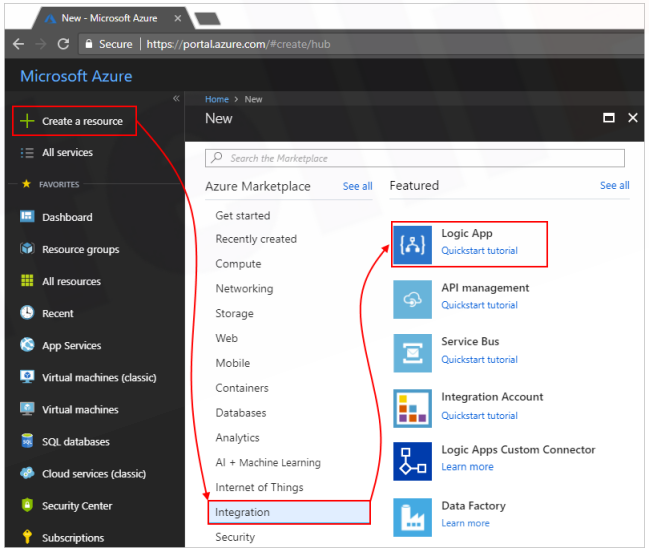
This logic app uses an Office 365 Outlook account. If you use another email account, the overall steps are the same, but your UI might slightly differ.



Create Logic Apps - Apps Portal

Create your logic app

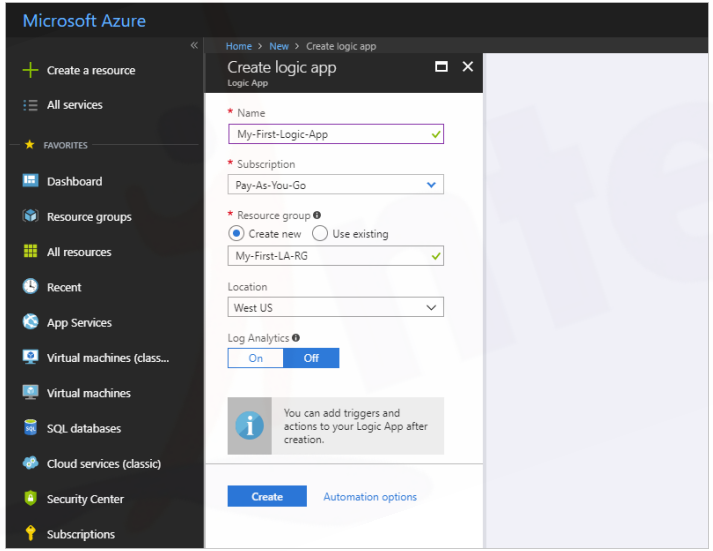
1. From the main Azure menu, choose Create a resource > Integration > Logic App.



Create Logic Apps - Apps Portal

Create your logic app

2. Under Create logic app, provide details about your logic app as shown here. After you're done, choose Create.

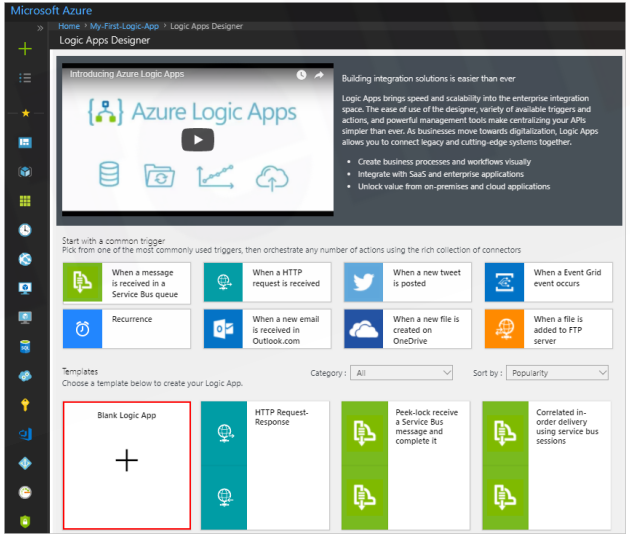


Property	Value	Description
Name	MyFirstLogicApp	The name for your logic app
Subscription	*<your-Azure-subscription-name>*	The name for your Azure subscription
Resource group	My-First-LA-RG	The name for the Azure resource group used to organize related resources
Location	West US	The region where to store your logic app information
Log Analytics	Off	Keep the **Off** setting for diagnostic logging.

Create Logic Apps - Apps Portal

Create your logic app

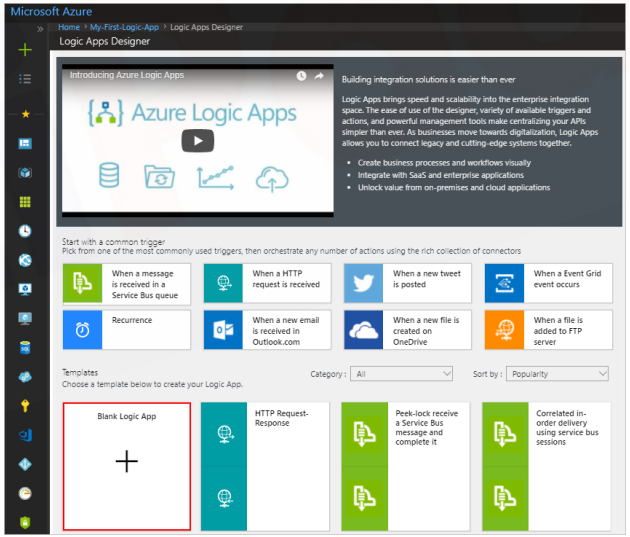
3. After Azure deploys your app, the Logic Apps Designer opens and shows a page with an introduction video and commonly used triggers. Under Templates, choose Blank Logic App.



Create Logic Apps - Apps Portal

Create your logic app

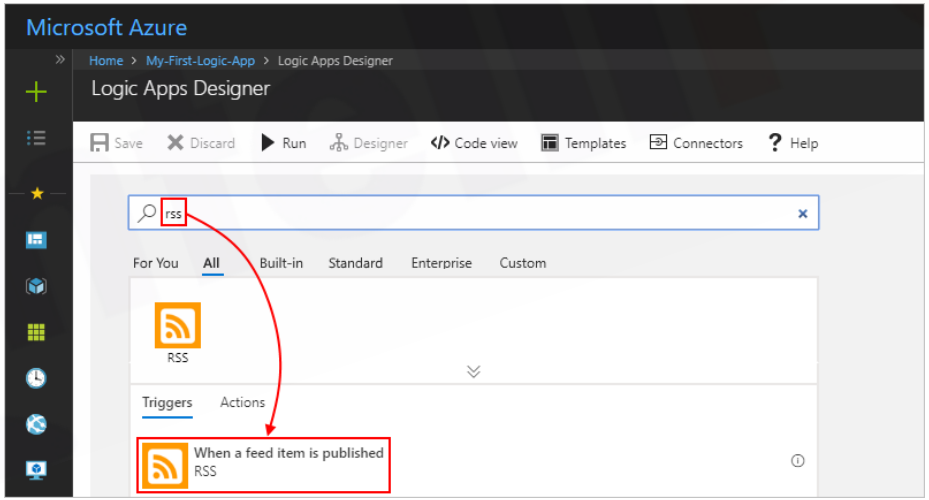
- Next, add a trigger that fires when a new RSS feed item appears. Every logic app must start with a trigger, which fires when a specific event happens or when a specific condition is met.
- Each time the trigger fires, the Logic Apps engine creates a logic app instance that starts and runs your workflow.



Create Logic Apps - Apps Portal

Check RSS feed with a trigger

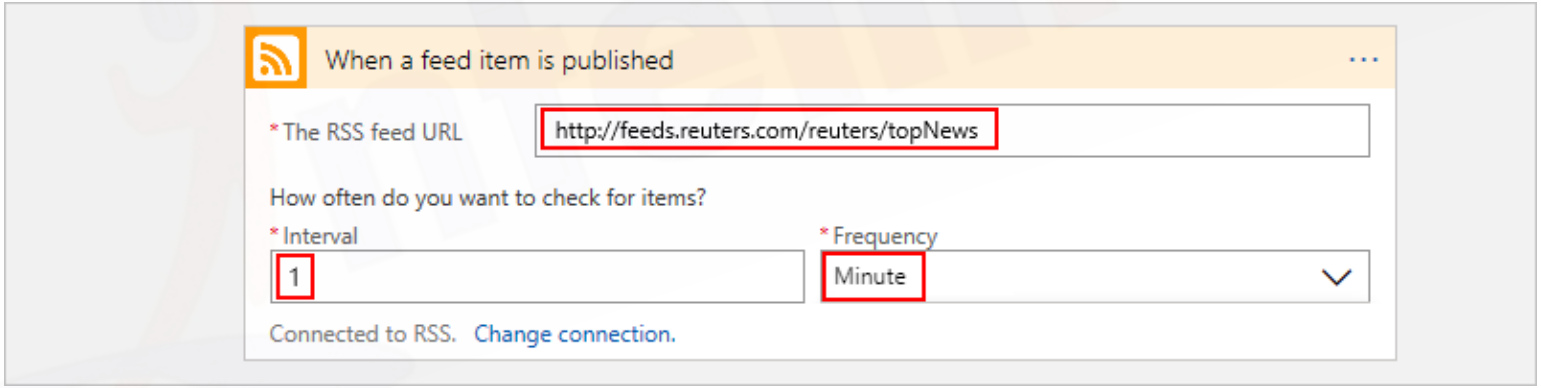
1. In Logic App Designer, under the search box, choose All.
2. In the search box, enter "rss". From the triggers list, select this trigger: When a feed item is published - RSS



Create Logic Apps - Apps Portal

Check RSS feed with a trigger

3. Provide this information for your trigger as shown and described:



Create Logic Apps - Apps Portal

Check RSS feed with a trigger

Property	Value	Description
The RSS feed URL	`http://feeds.reuters.com/reuters/topNews`	The link for the RSS feed that you want to monitor
Interval	1	The number of intervals to wait between checks
Frequency	Minute	The unit of time for each interval between checks

Together, the interval and frequency define the schedule for your logic app's trigger. This logic app checks the feed every minute.

Create Logic Apps - Apps Portal

4. To hide the trigger details for now, click inside the trigger's title bar.

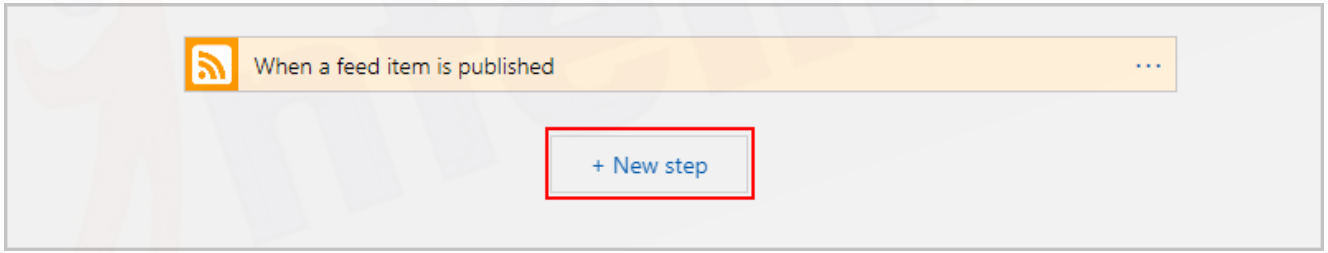
5. Save your logic app. On the designer toolbar, choose Save.
   Your logic app is now live but doesn't do anything other than check the RSS feed. So, add an action that responds when the trigger fires.

Create Logic Apps - Apps Portal

Send email with an action

Now add an action that sends email when a new item appears in the RSS feed.

1. Under the When a feed item is published trigger, choose New step.

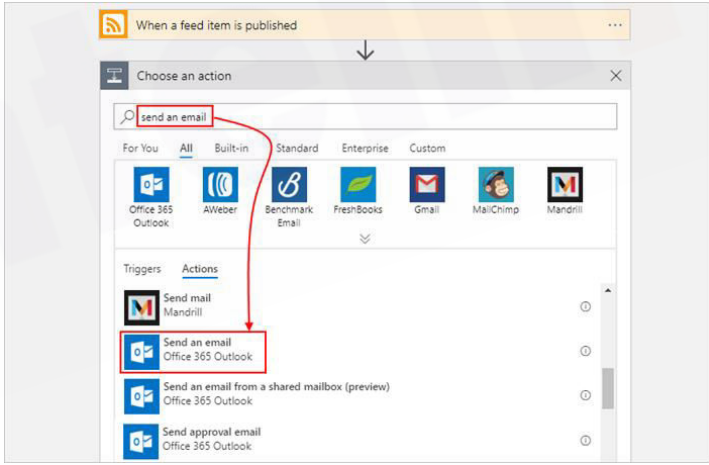


Create Logic Apps - Apps Portal

Send email with an action

2. Under Choose an action and the search box, choose All.

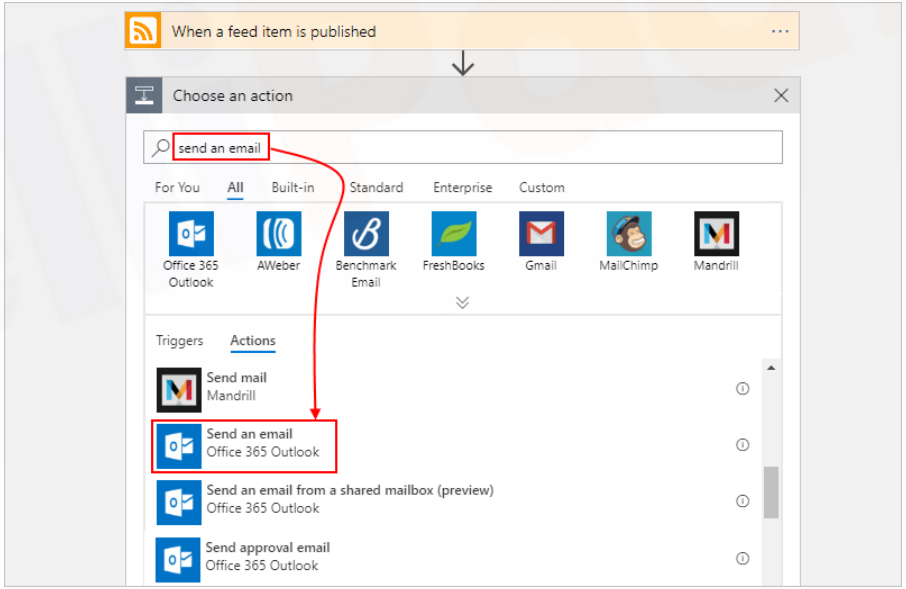
3. In the search box, enter "send an email". From the actions list, select the "send an email" action for the email provider that you want.



Create Logic Apps - Apps Portal

Send email with an action

- To filter the actions list to a specific app or service, you can select that app or service first:
  - For Azure work or school accounts, select Office 365 Outlook.
  - For personal Microsoft accounts, select Outlook.com.



Create Logic Apps - Apps Portal

Send email with an action

4. If asked for credentials, sign in to your email account so that Logic Apps creates a connection to your email account.

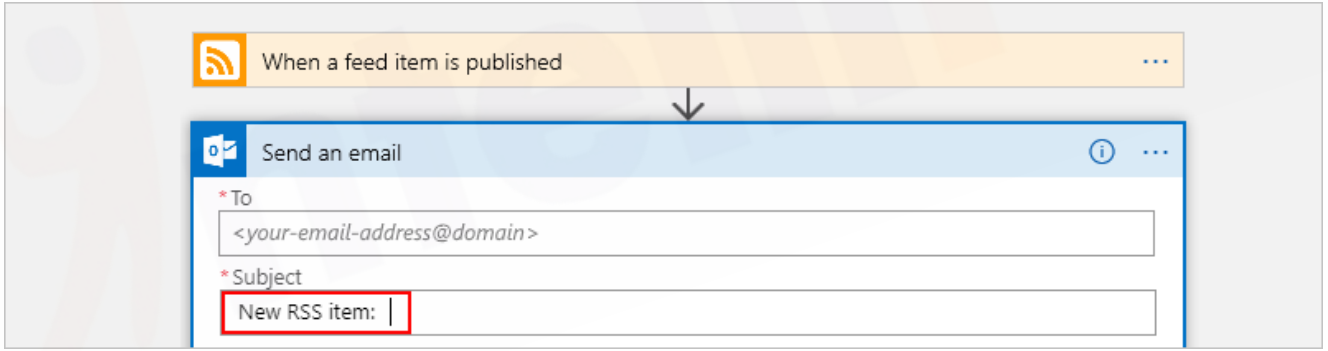
5. In the Send an email action, specify the data that you want the email to include:
    a. In the To box, enter the recipient's email address. For testing purposes, you can use your own email address.

For now, ignore the Add dynamic content list that appears. When you click inside some edit boxes, this list appears and shows any available parameters from the previous step that you can include as inputs in your workflow.

Create Logic Apps - Apps Portal

Send email with an action

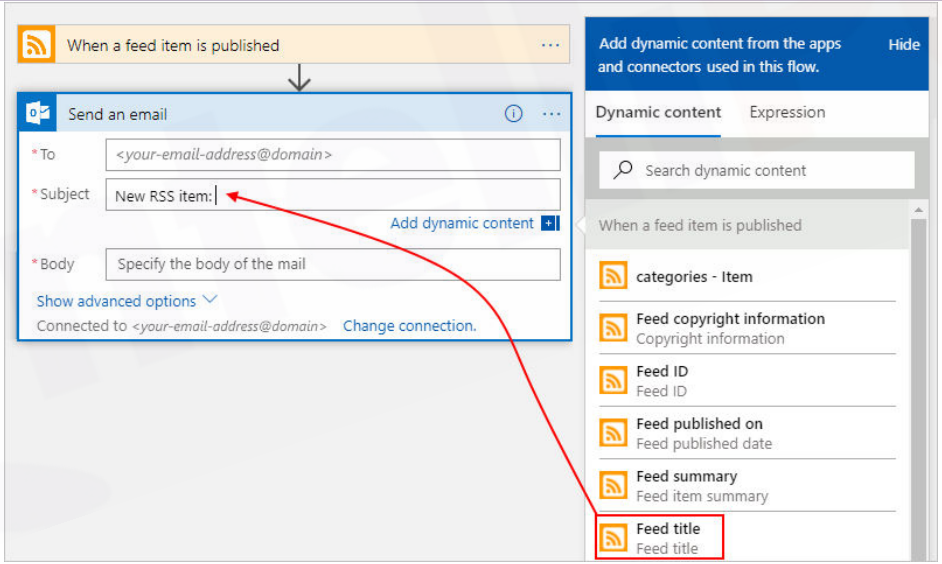
b. In the Subject box, enter this text with a trailing blank space: New RSS item:



Create Logic Apps - Apps Portal

Send email with an action

c. From the Add dynamic content list, select Feed title to include the RSS item title.

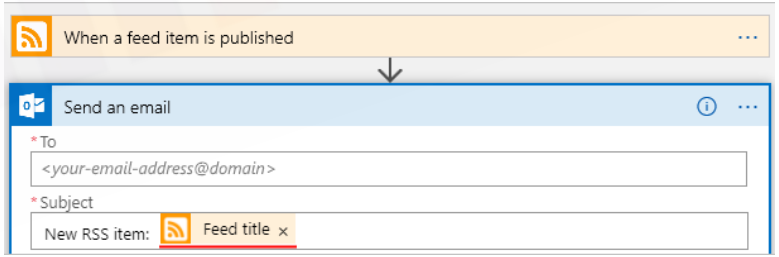


Create Logic Apps - Apps Portal

Send email with an action

- If a "For each" loop appears on the designer, then you selected a token for an array, for example, the categories-Item token.
- For these kinds of tokens, the designer automatically adds this loop around the action that references that token.
- That way, your logic app performs the same action on each array item.
- To remove the loop, choose the ellipses (...) on the loop's title bar, then choose Delete.

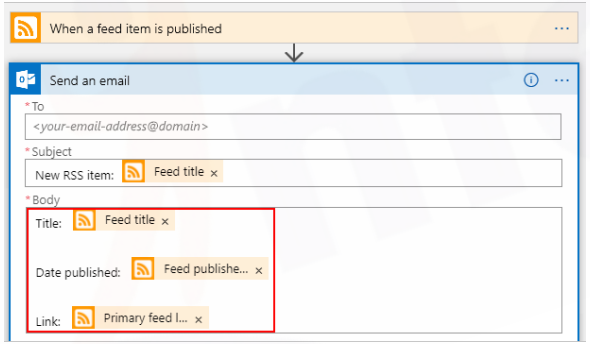
When you're done, the email subject looks like this example:



Create Logic Apps - Apps Portal

Send email with an action

d. In the Body box, enter this text, and select these tokens for the email body. To add blank lines in an edit box, press Shift + Enter.



Property	Description
Feed title	The item's title
Feed published on	The item's publishing date and time
Primary feed link	The URL for the item

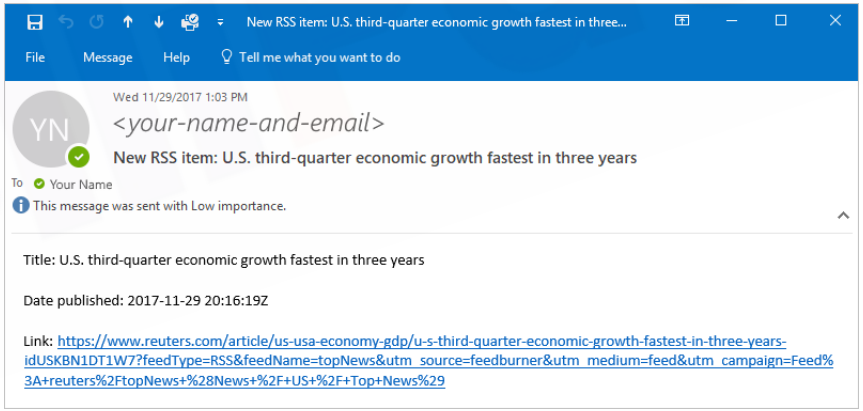
6. Save your logic app.

Create Logic Apps - Apps Portal

Run your logic app

- To manually start your logic app, on the designer toolbar bar, choose Run. Or, wait for your logic app to check the RSS feed based on your specified schedule (every minute).
- If the RSS feed has new items, your logic app sends an email for each new item. Otherwise, your logic app waits until the next interval before checking again.

For example, here is a sample email that this logic app sends. If you don't get any emails, check your junk email folder.

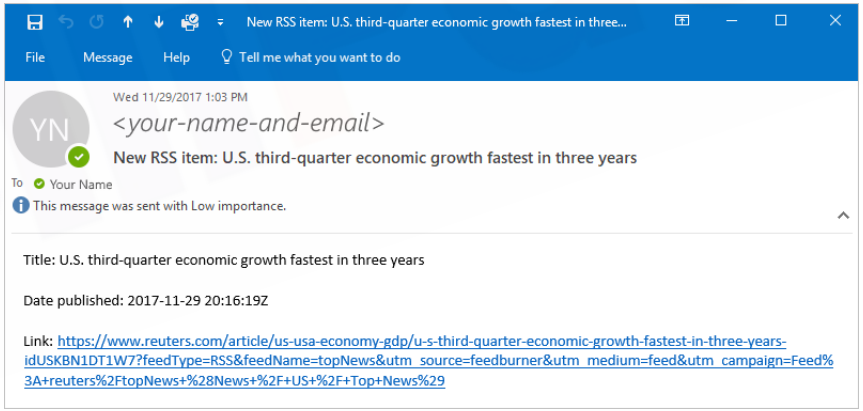


Create Logic Apps - Apps Portal

Run your logic app

- Technically, when the trigger checks the RSS feed and finds new items, the trigger fires, and the Logic Apps engine creates an instance of your logic app workflow that runs the actions in the workflow.
- If the trigger doesn't find new items, the trigger doesn't fire and "skips" instantiating the workflow.

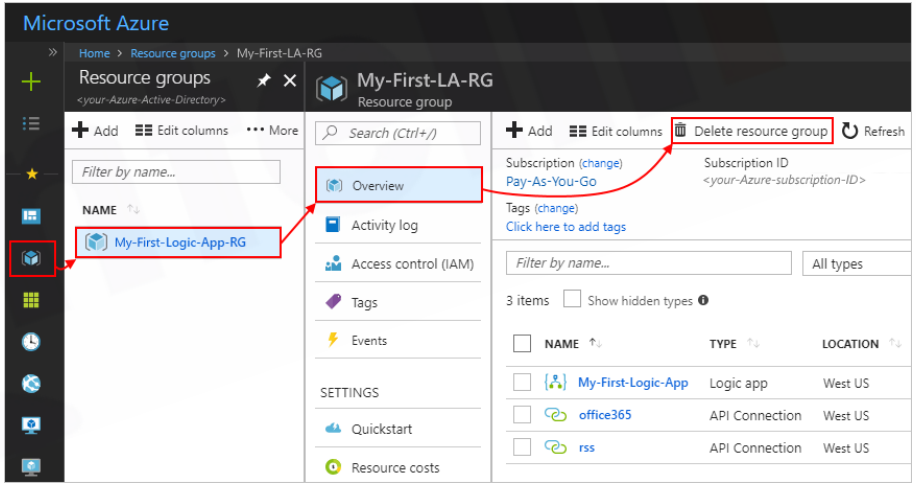
For example, here is a sample email that this logic app sends. If you don't get any emails, check your junk email folder.



Create Logic Apps - Apps Portal

Clean up resources

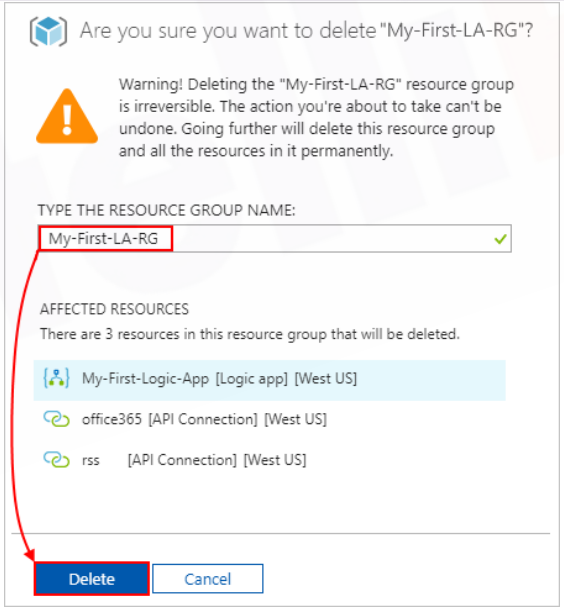
1. On the main Azure menu, go to Resource groups, and select your logic app's resource group. On the Overview page, choose Delete resource group.



Create Logic Apps - Apps Portal

Clean up resources

2. Enter the resource group name as confirmation, and choose Delete.



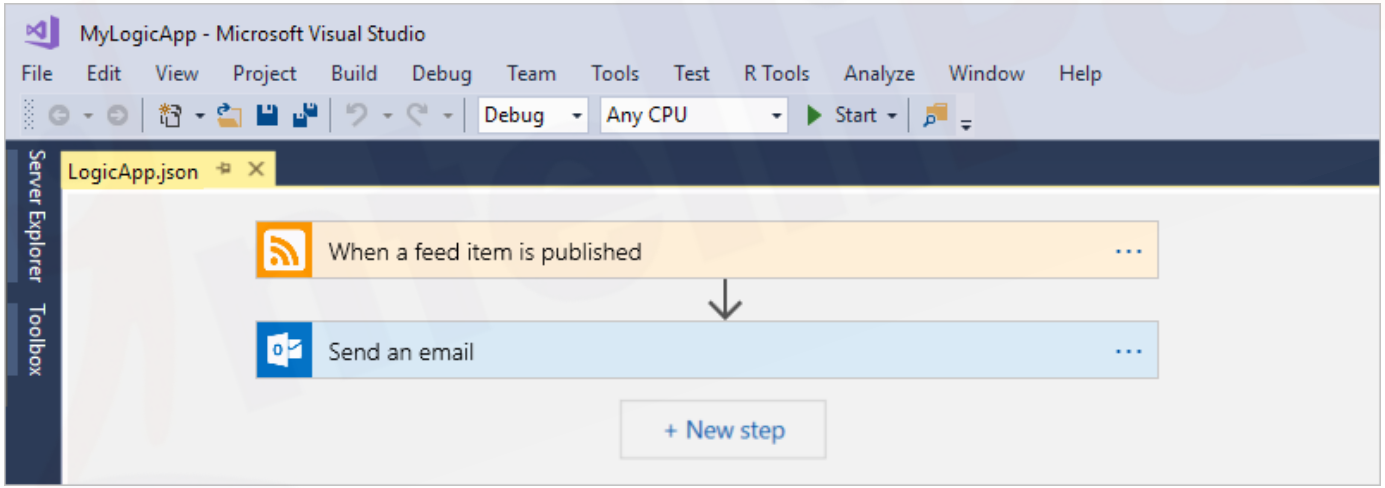
Create Logic Apps - Visual Studio

Create Logic Apps - Apps Studio

- With Azure Logic Apps and Visual Studio, you can create workflows for automating tasks and processes that integrate apps, data, systems, and services across enterprises and organizations.
- You can perform these tasks in the Azure portal, Visual Studio lets you add your logic apps to source control, publish different versions, and create Azure Resource Manager templates for different deployment environments.
- If you're new to Azure Logic Apps and just want the basic concepts. The Logic App Designer works similarly in both the Azure portal and Visual Studio.

Create Logic Apps - Apps Studio

This logic app monitors a website's RSS feed and sends email for each new item in that feed. Your finished logic app looks like this high-level workflow:



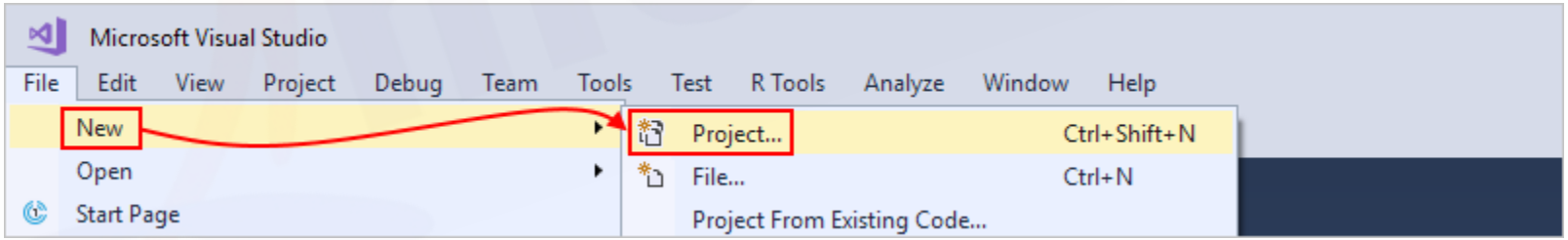
Create Logic Apps – Visual Studio

Create Azure resource group project

To get started, create an Azure Resource Group project. Learn more about Azure resource groups and resources.

1. Start Visual Studio. Sign in with your Azure account.

2. On the File menu, select New > Project. (Keyboard: Ctrl+Shift+N)

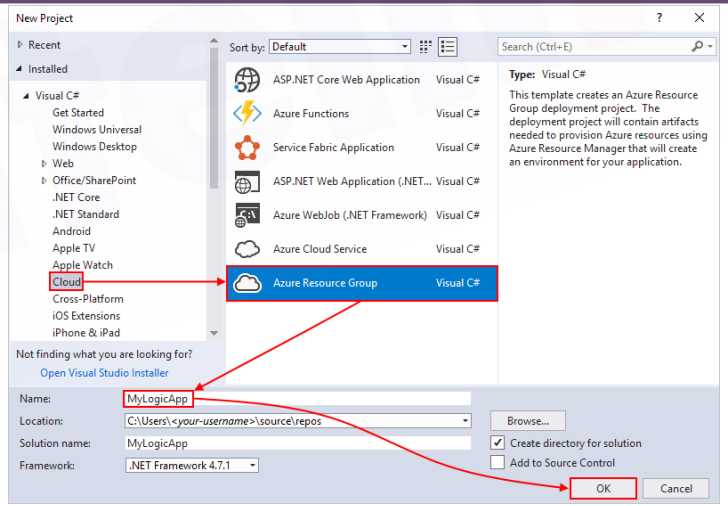


Create Logic Apps – Visual Studio

Create Azure resource group project

To get started, create an Azure Resource Group project. Learn more about Azure resource groups and resources.

3. Under Installed, select Visual C# or Visual Basic. Select Cloud > Azure Resource Group. Name your project, for example:



Create Logic Apps – Visual Studio

Create Azure resource group project

To get started, create an Azure Resource Group project. Learn more about Azure resource groups and resources.

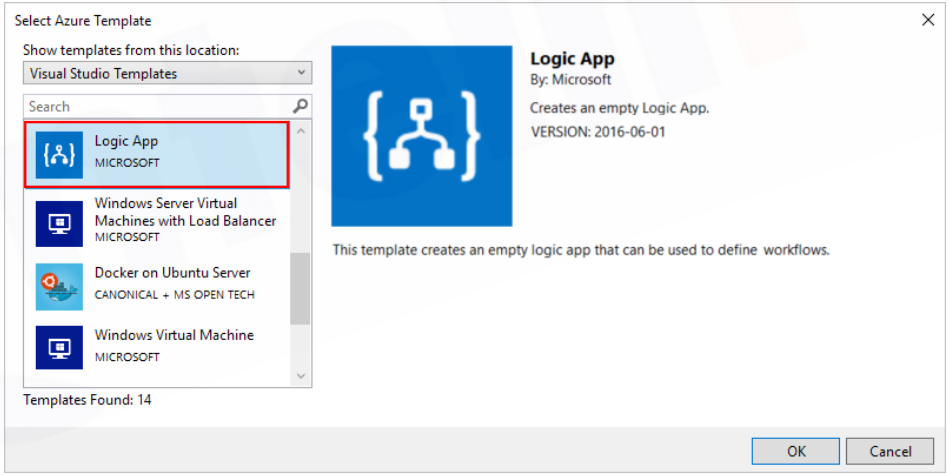
If you're using Visual Studio 2019, follow these steps:

a. In the Create a new project box, select the Azure Resource Group project for Visual C# or Visual Basic. Choose Next.
b. Provide a name for the Azure resource group you want to use and other project information. Choose Create.

Create Logic Apps – Visual Studio

Create Azure resource group project

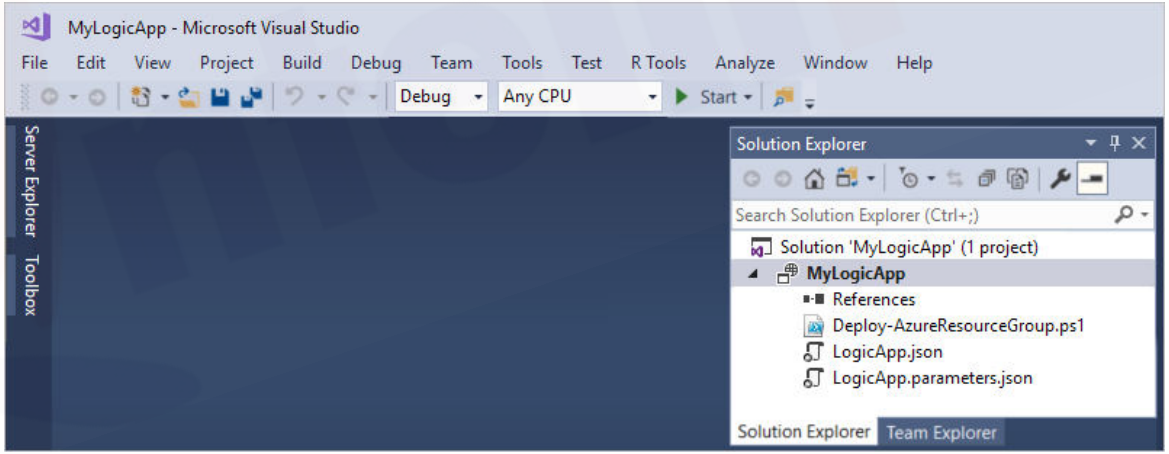
4. From the template list, select the LogicApp template. Choose OK.Provide a name for the Azure resource group you want to use and other project information. Choose Create.



Create Logic Apps – Visual Studio

Create Azure resource group project

After Visual Studio creates your project, Solution Explorer opens and shows your solution. In your solution, the LogicApp.json file not only stores your logic app definition but is also an Azure Resource Manager template that you can use for deployment.

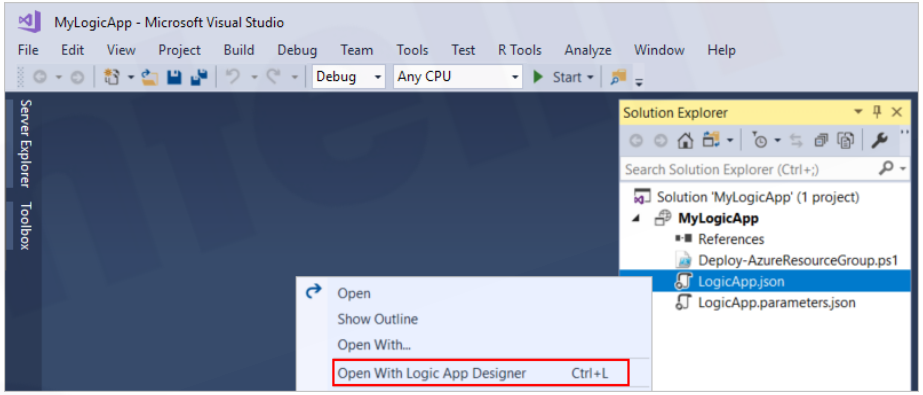


Create Logic Apps – Visual Studio

Create blank logic app

When you have your Azure Resource Group project, create your logic app with the Blank Logic App template.

1. In Solution Explorer, open the LogicApp.json file's shortcut menu. Select Open With Logic App Designer. (Keyboard: Ctrl+L)

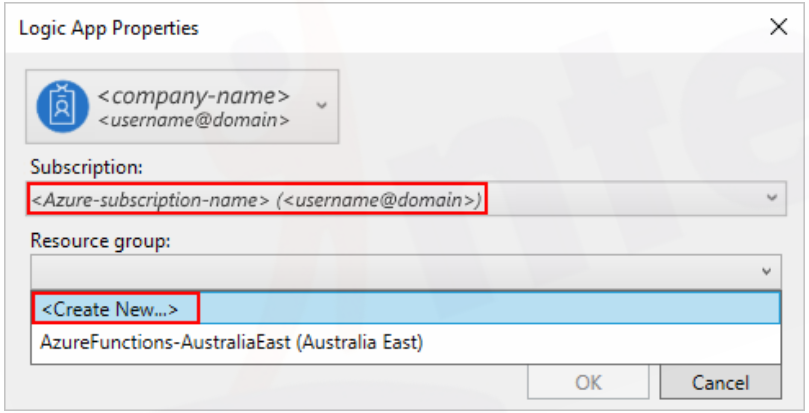


Visual Studio prompts you for your Azure subscription and an Azure resource group for creating and deploying resources for your logic app and connections.

Create Logic Apps – Visual Studio

Create blank logic app

2. For Subscription, select your Azure subscription. For Resource group, select Create New to create a new Azure resource group.

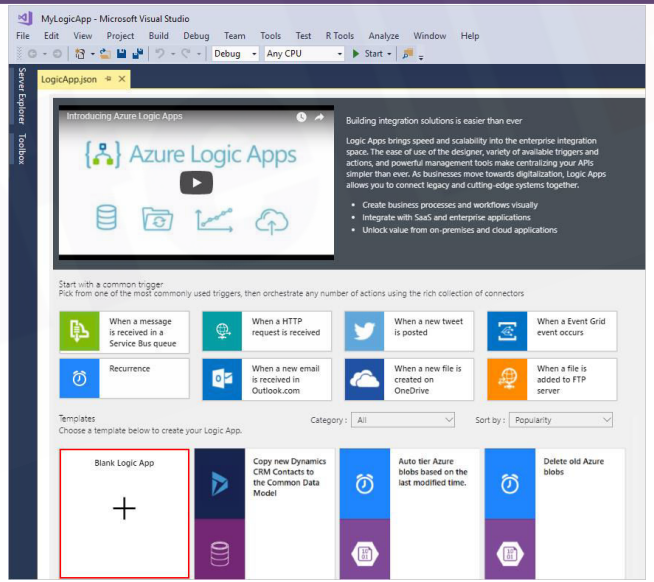


Setting	Example value	Description
User profile list	Contoso jamalhartnett@contoso.com	By default, the account that you used to sign in
Subscription	Pay-As-You-Go (jamalhartnett@contoso.com)	The name for your Azure subscription and associated account
Resource Group	MyLogicApp-RG (West US)	The Azure resource group and location for storing and deploying your logic app's resources
Location	MyLogicApp-RG2 (West US)	A different location if you don't want to use the resource group location

Create Logic Apps – Visual Studio

Create blank logic app

3. The Logic Apps Designer opens a page that shows an introduction video and commonly used triggers. Scroll down past the video and triggers to Templates, and select Blank Logic App.



Create Logic Apps – Visual Studio

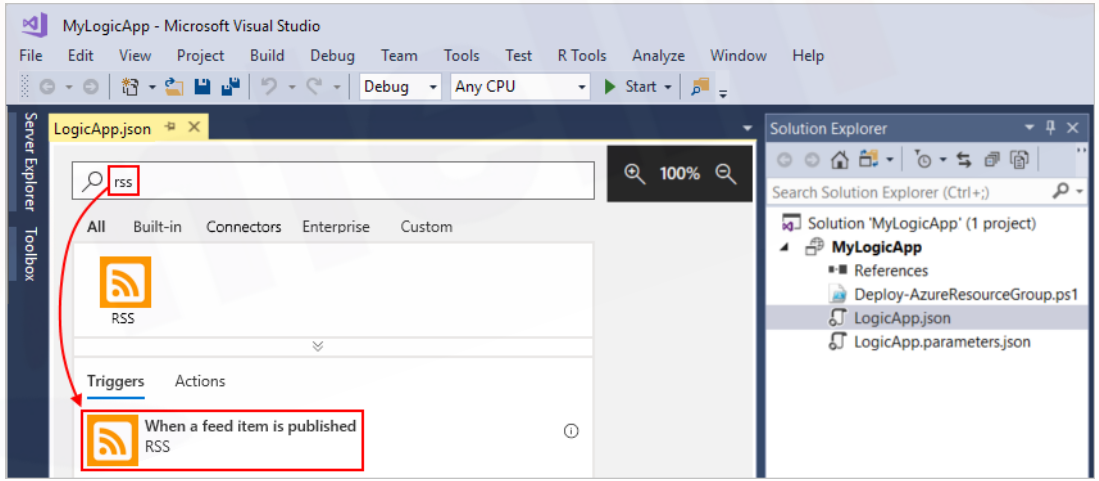
Build logic app workflow

- Next, add an RSS trigger that fires when a new feed item appears.
- Every logic app starts with a trigger, which fires when specific criteria is met.
- Each time the trigger fires, the Logic Apps engine creates a logic app instance that runs your workflow.

Create Logic Apps – Visual Studio

Build logic app workflow

1. In Logic App Designer, under the search box, choose All. In the search box, enter "rss". From the triggers list, select this trigger: When a feed item is published - RSS

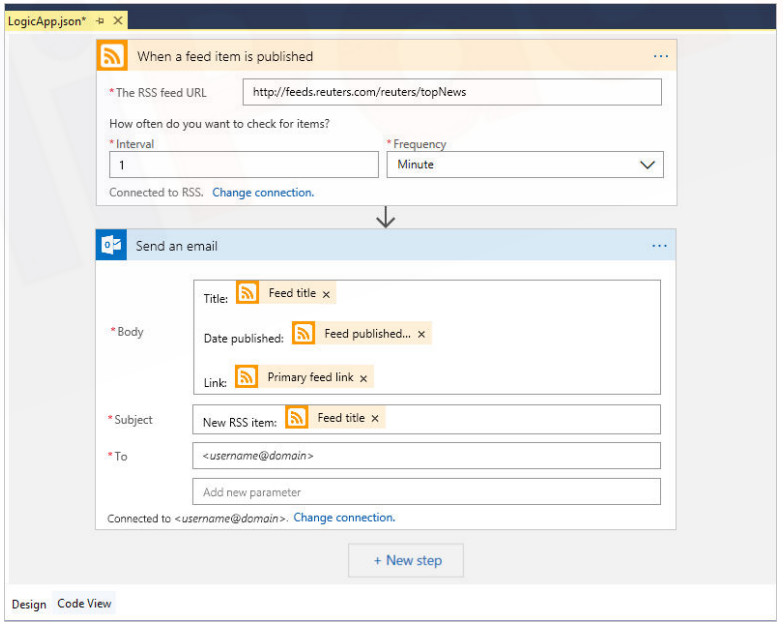


Create Logic Apps – Visual Studio

Build logic app workflow

2. After the trigger appears in the designer, finish building the logic app by following the workflow steps in the Azure portal quickstart, then return to this article.
3. Save your Visual Studio solution. (Keyboard: Ctrl + S)

When you're done, your logic app looks like this example:

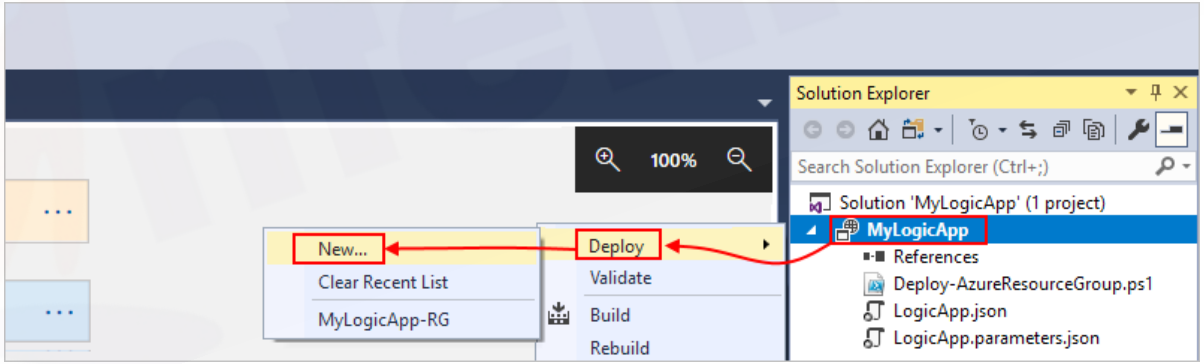


Create Logic Apps – Visual Studio

Deploy logic app to Azure

Before you can run and test your logic app, deploy the app to Azure from Visual Studio:

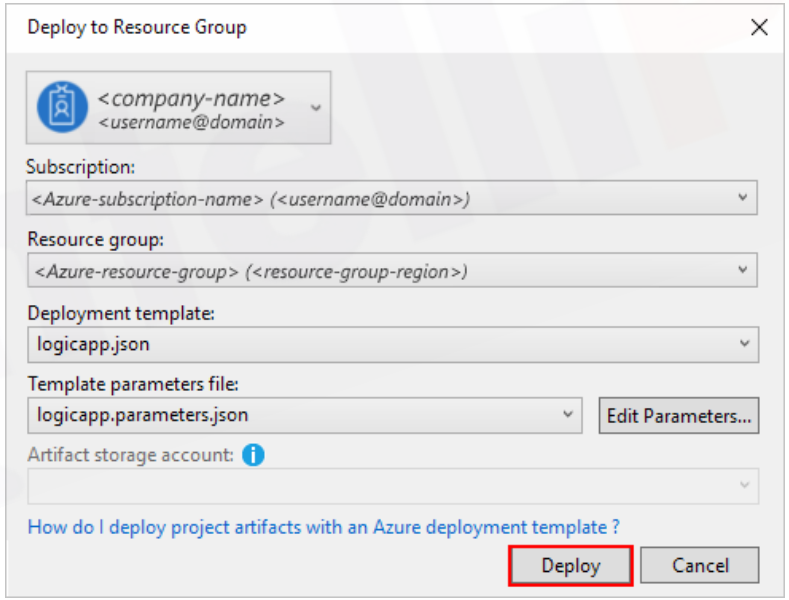
1. In Solution Explorer, on your project's shortcut menu, select Deploy > New. If prompted, sign in with your Azure account.Save your Visual Studio solution. (Keyboard: Ctrl + S)



Create Logic Apps – Visual Studio

Deploy logic app to Azure

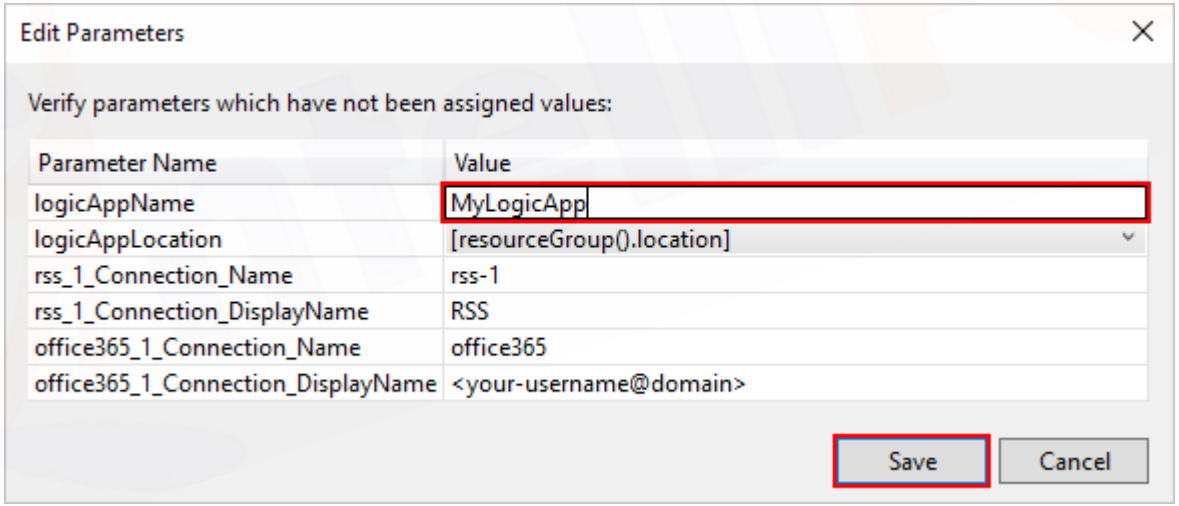
2. For this deployment, keep the default Azure subscription, resource group, and other settings. Choose Deploy.



Create Logic Apps – Visual Studio

Deploy logic app to Azure

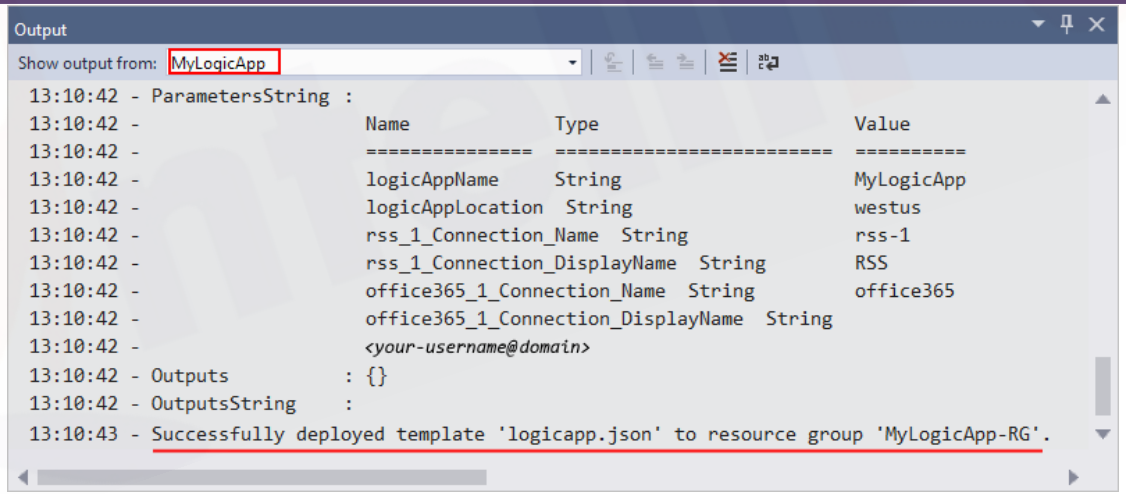
3. If the Edit Parameters box appears, provide a resource name for your logic app. Save your settings.



Create Logic Apps – Visual Studio

Deploy logic app to Azure

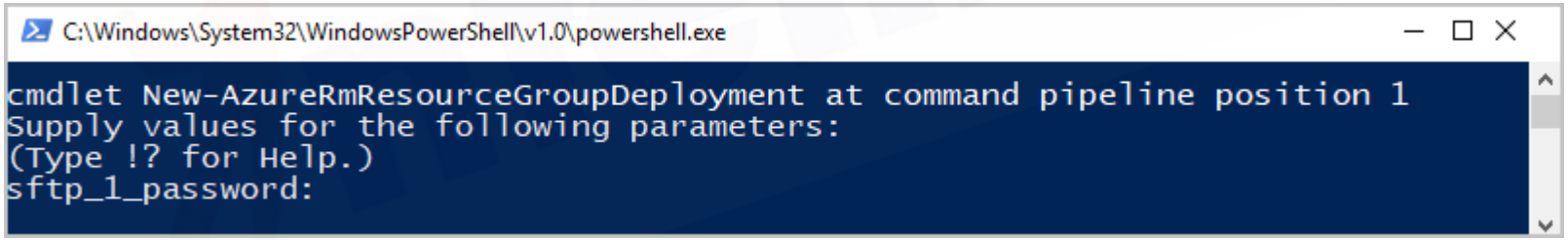
When deployment starts, your app's deployment status appears in the Visual Studio Output window. If the status doesn't appear, open the Show output from list, and select your Azure resource group.



Create Logic Apps – Visual Studio

Deploy logic app to Azure

If your selected connectors need input from you, a PowerShell window opens in the background and prompts for any necessary passwords or secret keys. After you enter this information, deployment continues.

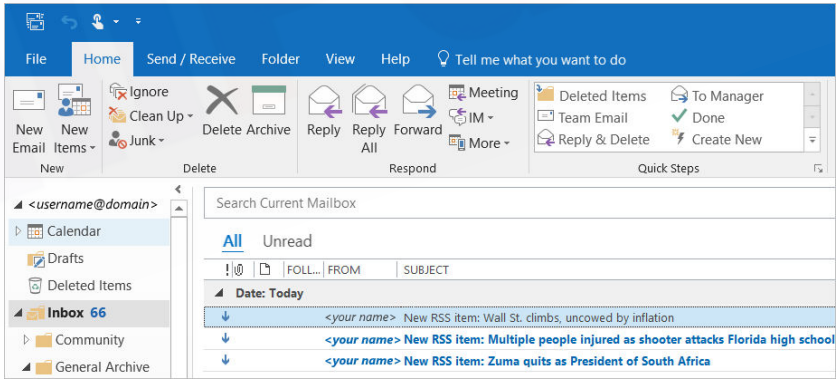


Create Logic Apps – Visual Studio

Deploy logic app to Azure

- After deployment finishes, your logic app is live in the Azure portal and runs on your specified schedule (every minute).
- If the trigger finds new feed items, the trigger fires, which creates a workflow instance that runs your logic app's actions.
- Your logic app sends email for each new item. Otherwise, if the trigger doesn't find new items, the trigger doesn't fire and "skips" instantiating the workflow. Your logic app waits until the next interval before checking.

Here are sample emails that this logic app sends. If you don't get any emails, check your junk email folder.

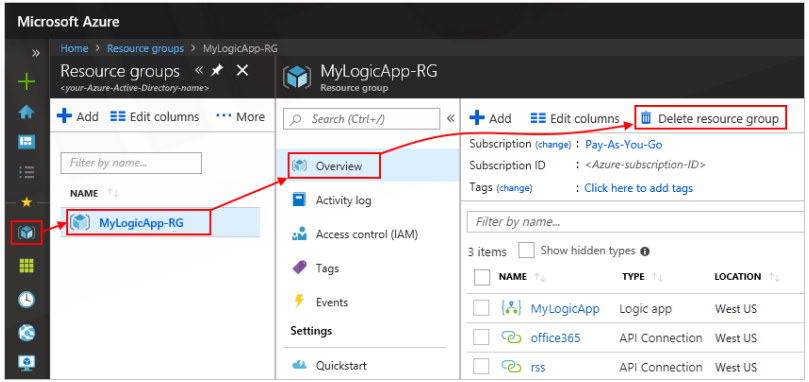


Create Logic Apps – Visual Studio

Clean up resources

When you're done with your logic app, delete the resource group that contains your logic app and related resources.

1. Sign in to the Azure portal with the same account used to create your logic app.
2. On the main Azure menu, select Resource groups. Select your logic app's resource group, and select Overview.
3. On the Overview page, choose Delete resource group. Enter the resource group name as confirmation, and choose Delete.
4. Delete the Visual Studio solution from your local computer.



Connectors

Connectors

- A connector is a proxy or a wrapper around an API that allows the underlying service to talk to Microsoft Flow, PowerApps and Logic Apps.
- It provides a way for users to connect their accounts and leverage a set of pre-built actions and triggers to build their apps and workflows.

Components of a Connector

- Each connector offers a set of operations classified as '**Actions**' and '**Triggers**'.
- Once you connect to the underlying service, these operations can be easily leveraged within your apps and workflows.

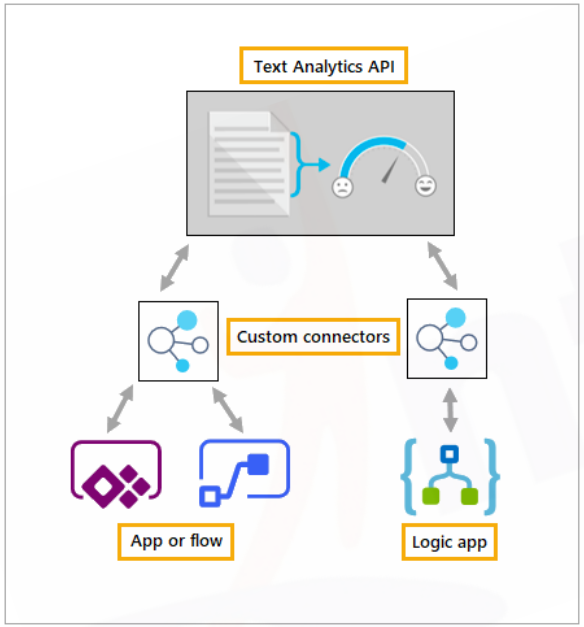
Actions
- Actions are changes directed by user, look up, write, update, or delete data.

Triggers
- Triggers that notify your app during specific events.

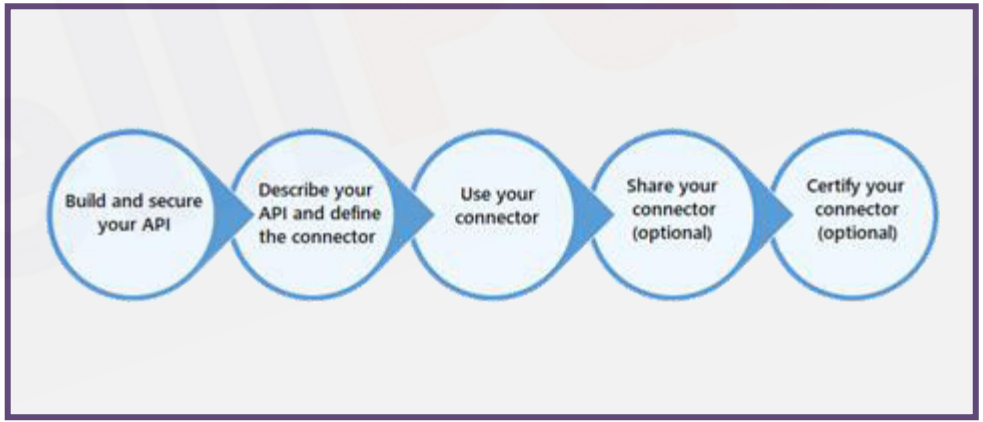
Custom Connectors

Custom Connectors

- Custom connectors address this scenario by allowing you to create (and even share) a connector with its own triggers and actions.



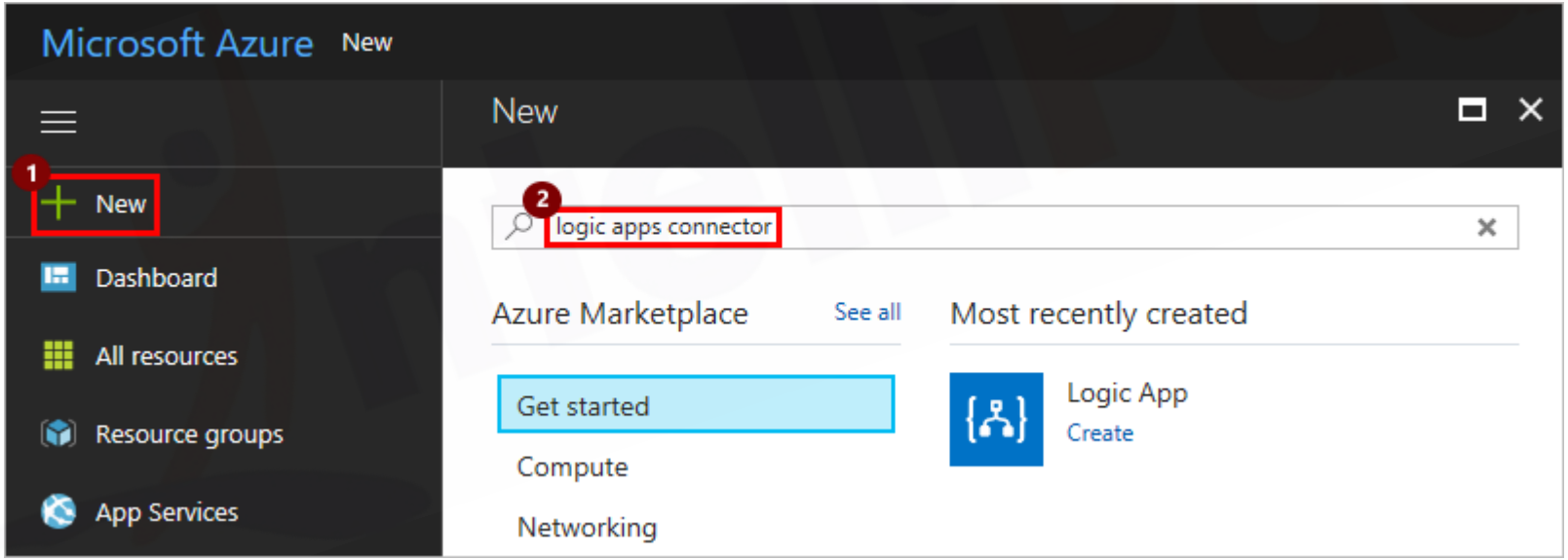
Lifecycle



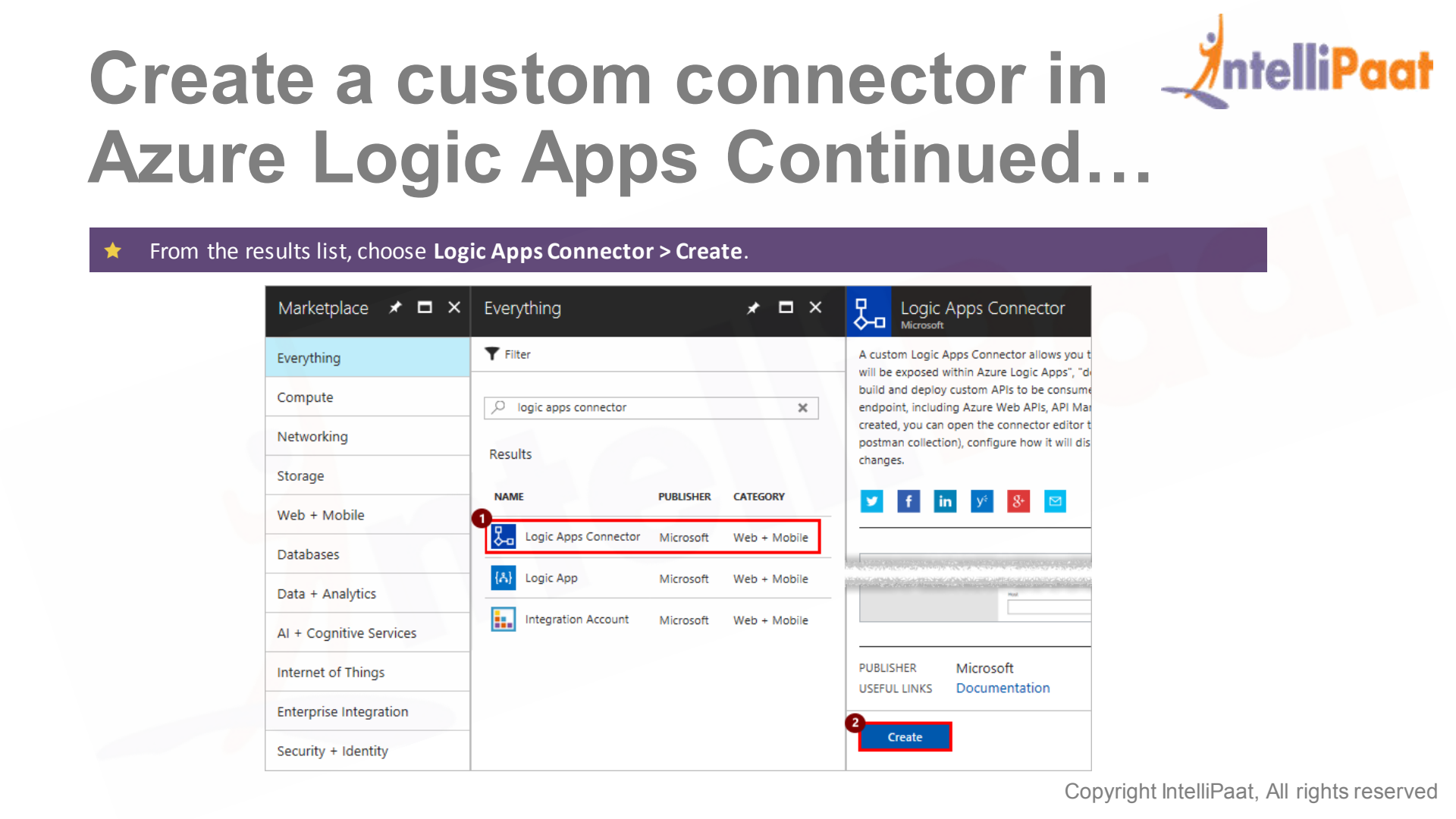
Create a custom connector in Azure Logic Apps

Create a custom connector in Azure Logic Apps

- In the Azure portal, on the main Azure menu, choose New. In the search box, enter "logic apps connector" as your filter, and press Enter.



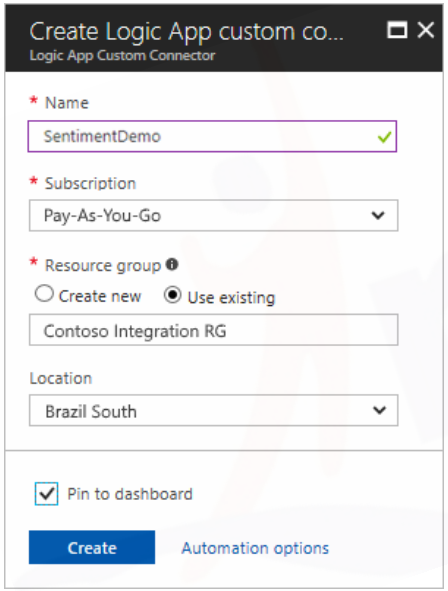
Create a custom connector in Azure Logic Apps Continued…



★ From the results list, choose **Logic Apps Connector > Create**.

Create a custom connector in Azure Logic Apps Continued…

★ Provide details for registering your connector as described in the table. When you're done, choose **Pin to dashboard > Create**.



Property	Description	Example
Name	Name of your custom connector	"SentimentDemo"
Subscription	Azure subscription for the connector	"Contoso Azure Subscription #1"
Resource group	Azure resource group for the connector	"logic-apps-custom-connectors"
Location	Azure region	"West US"

Hands-on

Hands-on

- **Create logic app workflows from prebuilt templates**
  - Create logic apps from templates
  - Update logic apps with templates
  - Deploy logic apps built from templates

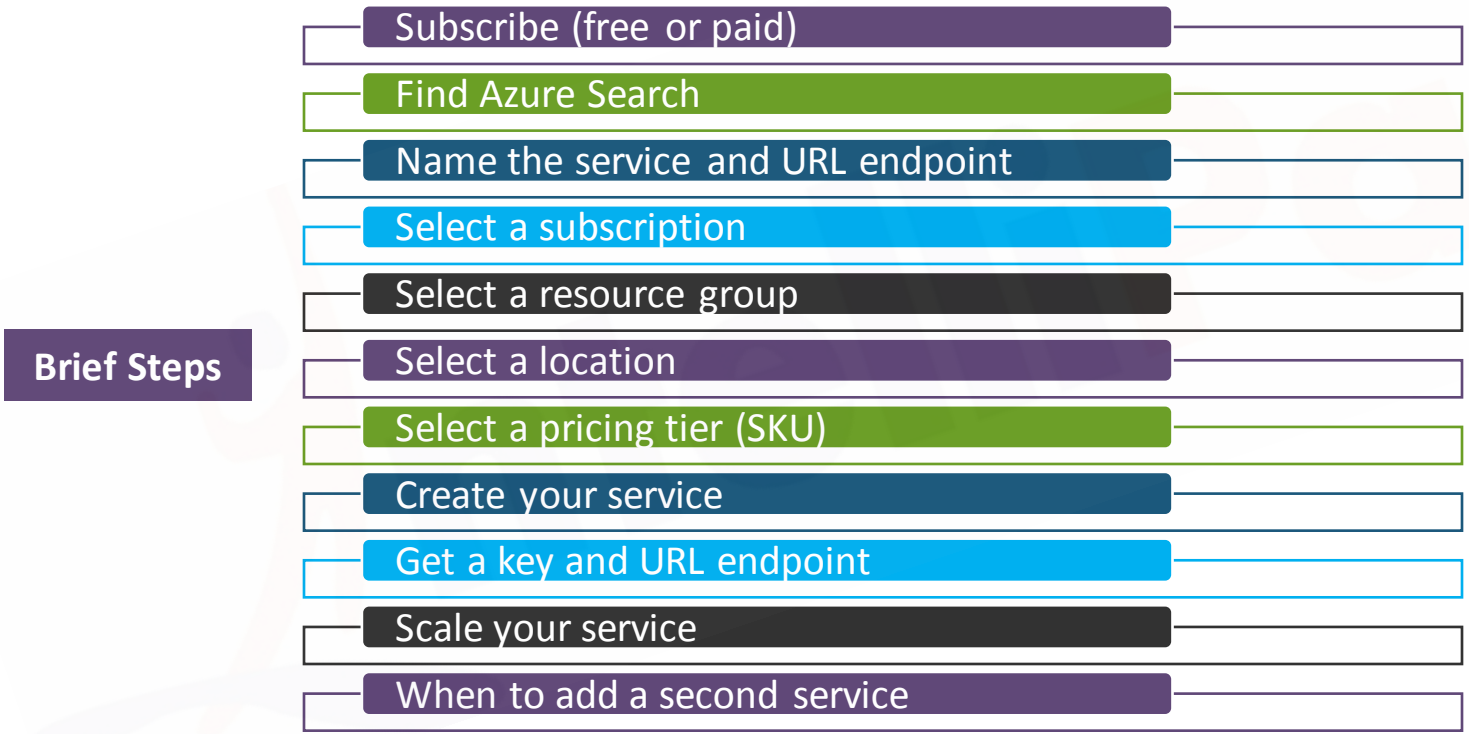
Azure Search

What is Azure Search?

- Azure Search is a search-as-a-service cloud solution that gives developers APIs and tools for adding a rich search experience over private, heterogenous content in web, mobile, and enterprise applications. Query execution is over a user-defined index.

- Build a search index containing only your data, sourced from multiple content types and platforms.
- Leverage AI enrichments to extract text and features from image files, or entities and key phrases from raw text.
- Create intuitive search experiences with facet navigation and filters, synonyms, autocomplete, and text analysis for "did you mean" autocorrected search terms. Get relevance tuning through functions and boosting logic.
- Create search apps for specific use-cases. Geo-search supports a "find near me" experience. Multi-lingual search is supported through language analysers for non-English full text search.

How to create an Azure Search Service in the Portal?



Creating an Azure Search INDEX in the Portal

- Azure Search includes a built-in index designer in the portal useful for prototypes or creating a search index hosted on your Azure Search service.
- The tool is used for schema construction. When you save the definition, an empty index becomes fully expressed in Azure Search.

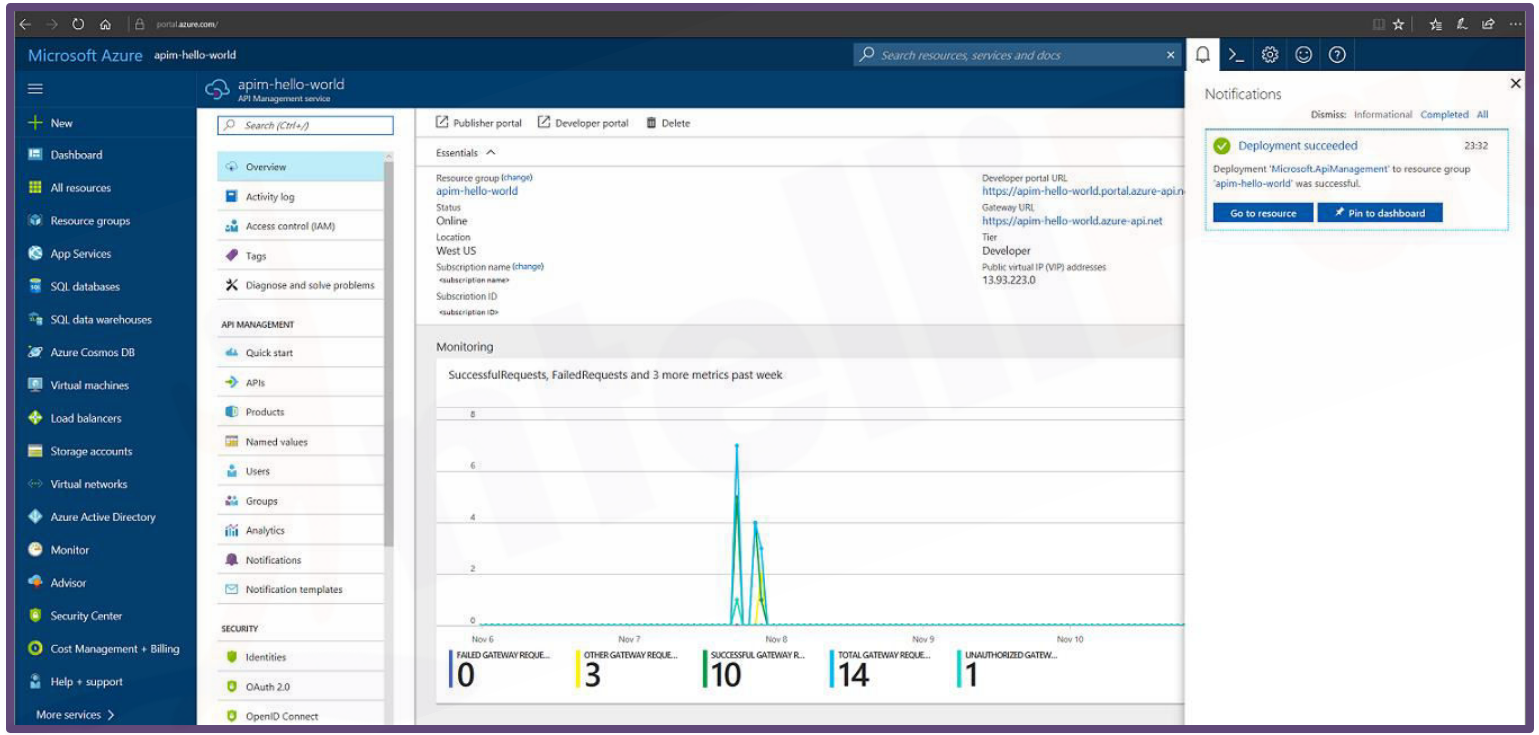
Brief Steps

- Start index designer
- Add fields
- Set attributes

Azure API Management

API Management (APIM)

★ API Management (APIM) is a way to create consistent and modern API gateways for existing back-end services.

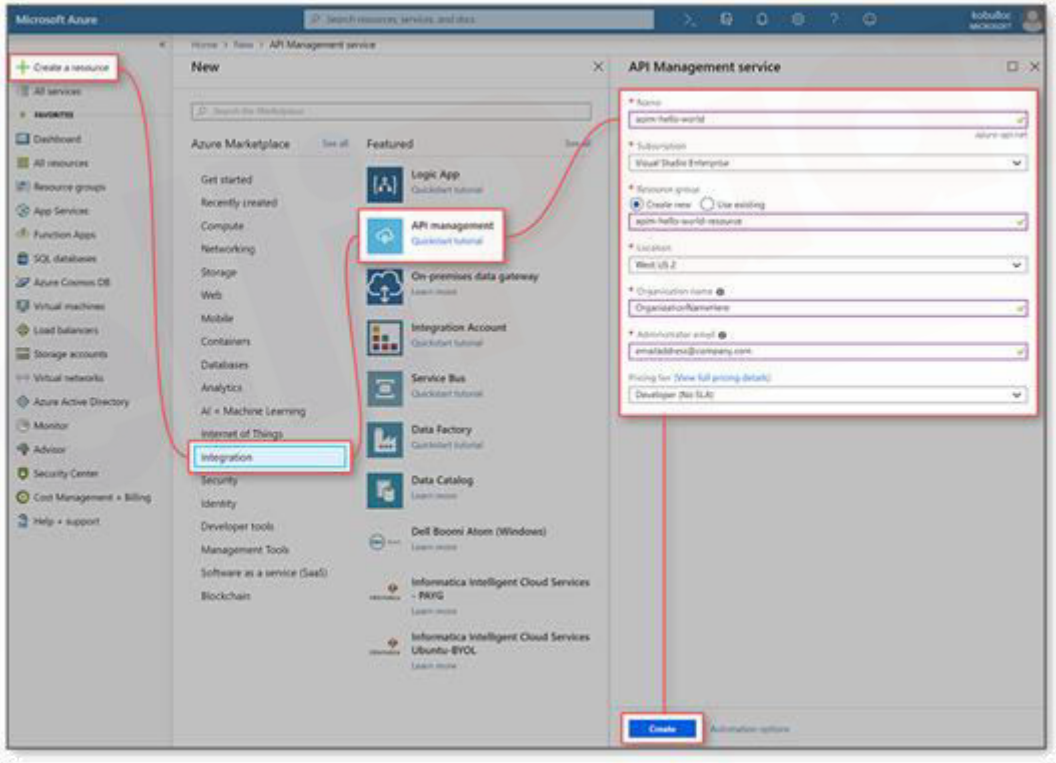


Create a new Azure API Management service instance

Creating a new Azure API Management service instance

Step 1

- ★ Login to Azure
- ★ Create a new service
- ★ In the Azure portal, select **Create a resource > Enterprise Integration > API management**.
- ★ Alternatively, choose **New**, **type API management in the search box**, and press **Enter**.
- ★ Click **Create**.

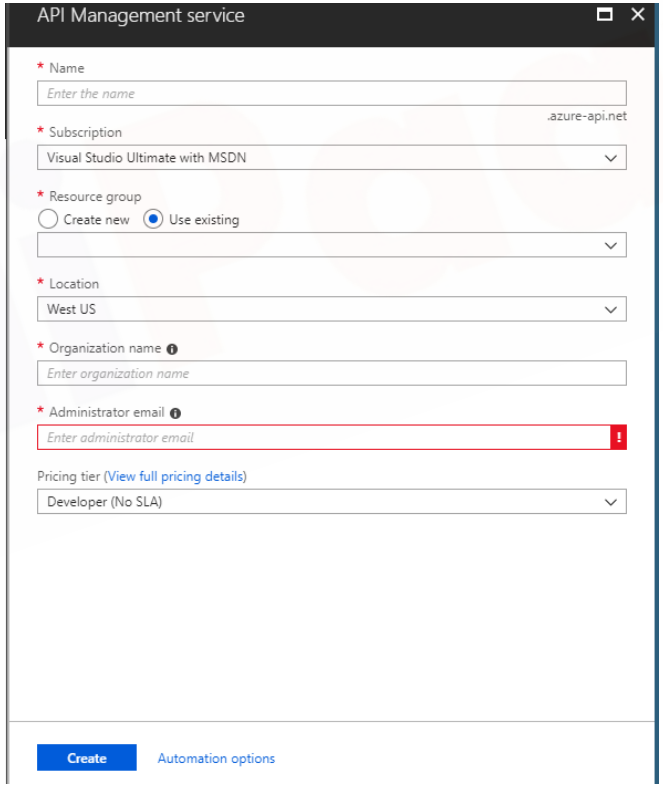


Creating a new Azure API Management service instance

Step 1 Continued….

- In the API Management service window, enter settings.

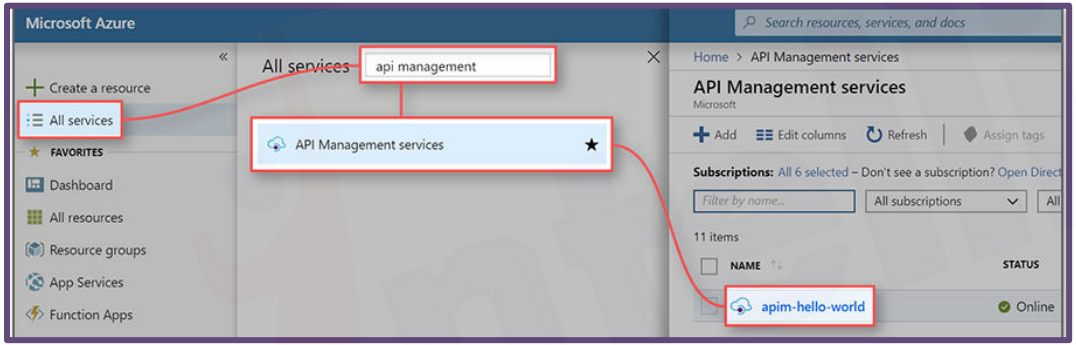
- Choose **Create**



Creating a new Azure API Management service instance

Step 2

★ Go to your API Management instance

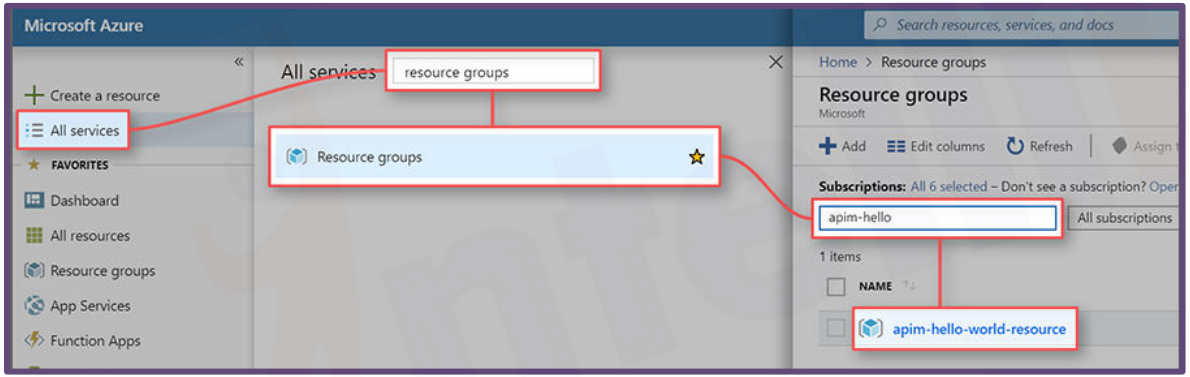


- ★ Sign in to the Azure portal.
- ★ Select All services.
- ★ In the search box, enter *api management*.
- ★ In the search results, select API Management services.
- ★ Select your API Management service instance.

Creating a new Azure API Management service instance

Last Step

★ When no longer needed, you can remove the resource group and all related resources by following these steps:

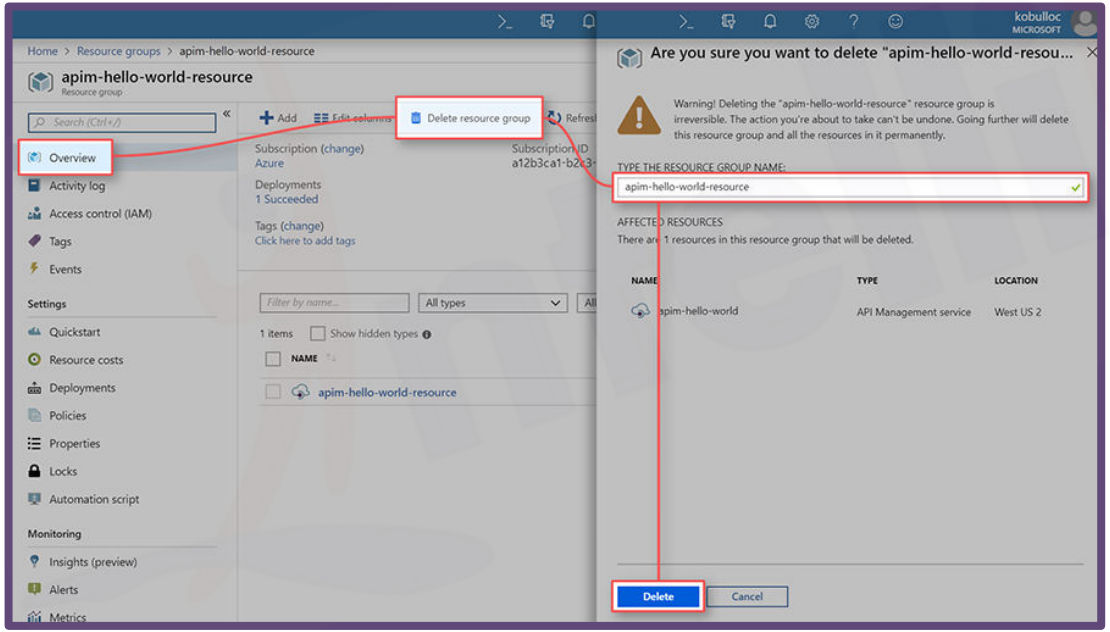


- ★ In the Azure portal, select All services.
- ★ Input resource groups in the search box and click on the result.

Creating a new Azure API Management service instance

Last Step Continued….

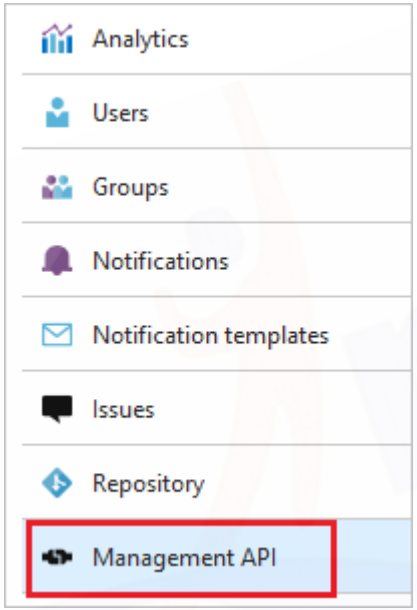
★ When no longer needed, you can remove the resource group and all related resources by following these steps:



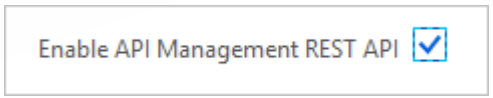
- ★ Find your resource group and click on it.
- ★ Click **Delete resource group**.
- ★ Confirm the deletion by inputting the *name of your resource group*.
- ★ Click **Delete**.

Then, authenticate your API

- We must use access token to in order to make calls into the Azure API Management REST API
- Manually create a SAS token



- Navigate to your Azure API Management instance in the Azure portal.
- Click Management API from the API Management section of the menu on the left.
- Make sure the Enable API Management REST API checkbox is selected.

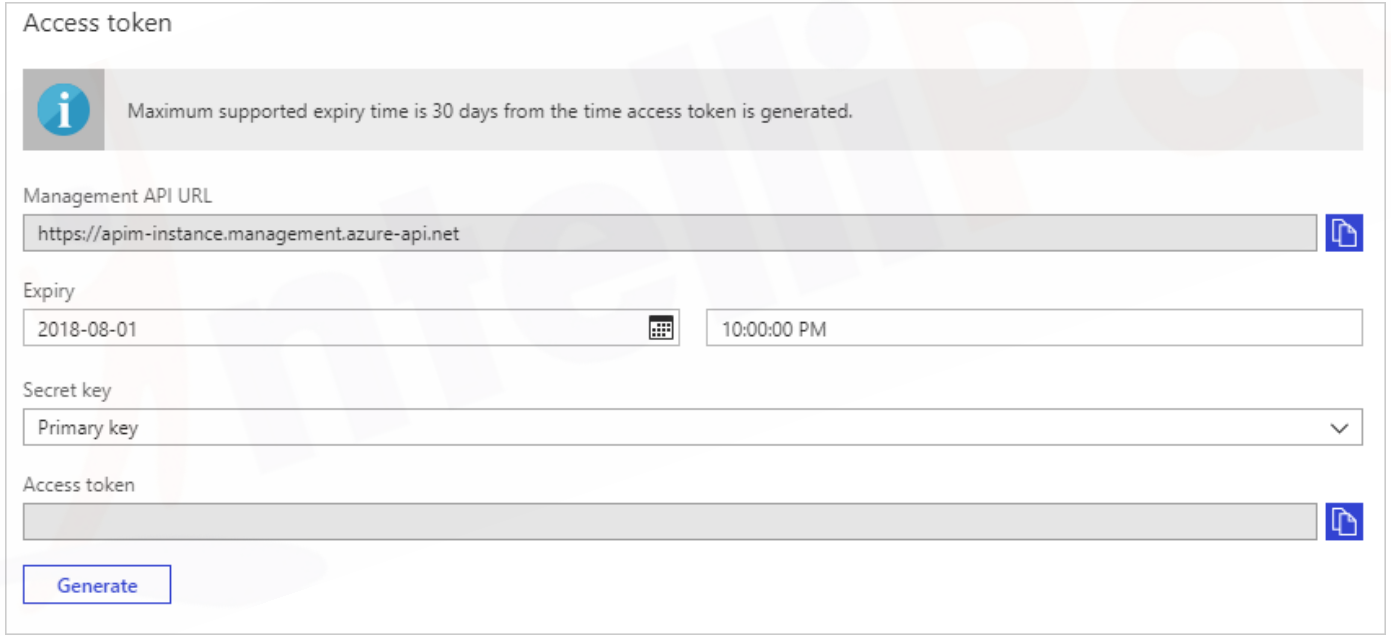


Then, authenticate your API

Further

- **Specify the expiration date and time for the access token in the Expiry text box.**

  This value must be in the format MM/DD/YYYY H:MM PM|AM.



Then, authenticate your API

Further

- Select either the primary key or secondary key in the Secret key drop-down list. The keys provide equivalent access; two keys are provided to enable flexible key management strategies.
- Click Generate to create the access token.
- Copy the full access token and provide it in the *Authorization* header of every request to the API Management REST API, as shown in the following example.

```
Authorization: SharedAccessSignature integration&201808020500&aAsTE43MAbKMkZ6q83Z732IbzesfsaPEU404oUjQ4ZLE9iIXLz+Jj9rEctxKYw43SioCfdLaDq7dT8RQuBKc0w==
```

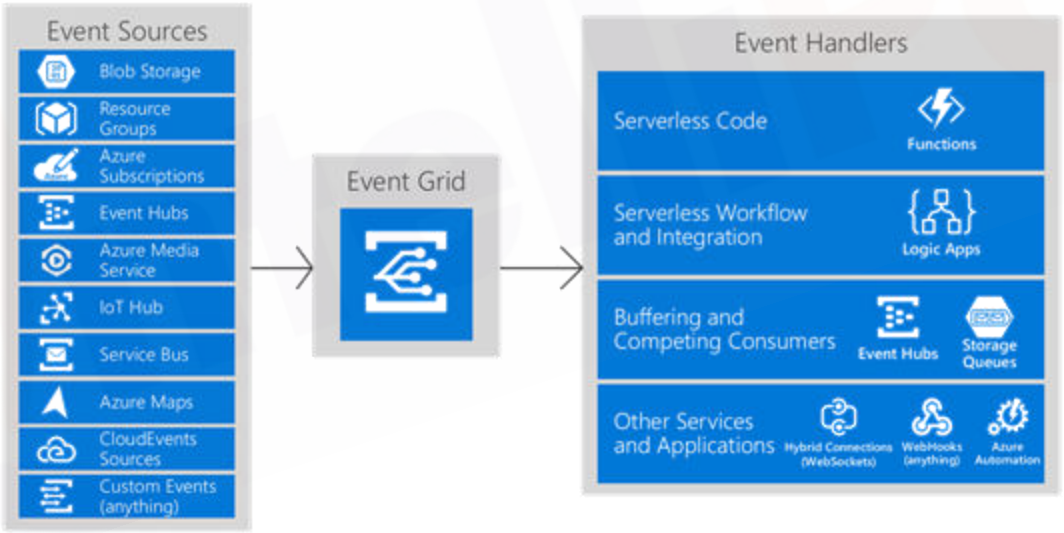
After this

- Programmatically create a SAS token

Azure Event Grid

What is Azure Event Grid?

- Azure Event Grid allows you to easily build applications with event-based architectures.
- Event Grid has built-in support for events coming from Azure services, like storage blobs and resource groups.
- Event Grid also has support for your own events, using custom topics.



How to Route Custom Events to Web Endpoint with Azure CLI and Event Grid

Route Custom Events to Web Endpoint with Azure CLI and Event Grid

Brief Steps

- First, open you Azure CLI
- Create a resource group
- Enable Event Grid resource provider
- Create a custom topic
- Create a message endpoint
- Subscribe to a custom topic
- Send an event to your custom topic
- Clean up resources

Route Custom Events to Web Endpoint with Azure CLI and Event Grid

Brief Steps

- First, open you Azure CLI
- Create a resource group
- Enable Event Grid resource provider
- Create a custom topic
- Create a message endpoint
- Subscribe to a custom topic
- Send an event to your custom topic
- Clean up resources

You can perform the similar steps in Azure Portal, as well as Azure Cloud Shell.

Azure Notification Hub

What is Azure Notification Hub?

- Azure Notification Hubs provide an easy-to-use and scaled-out push engine that allows you to send notifications to any platform (iOS, Android, Windows, Kindle, Baidu, etc.) from any backend (cloud or on-premises).
- Notification Hubs works great for both enterprise and consumer scenarios. Here are a few example scenarios:
  - Send breaking news notifications to millions with low latency.
  - Send location-based coupons to interested user segments.
  - Send event-related notifications to users or groups for media/sports/finance/gaming applications.
  - Push promotional contents to applications to engage and market to customers.
  - Notify users of enterprise events like new messages and work items.
  - Send codes for multi-factor authentication.

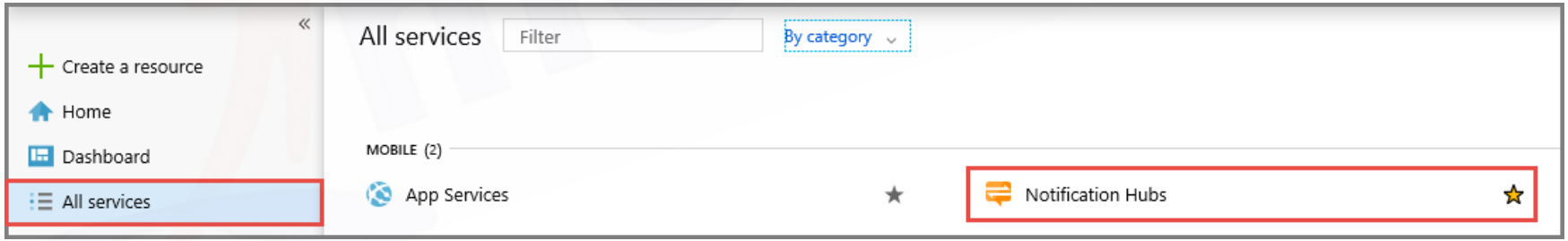
Create an Azure notification hub in the Azure portal

Create an Azure notification hub in the Azure portal

Step 1

★ Create a namespace and a notification hub

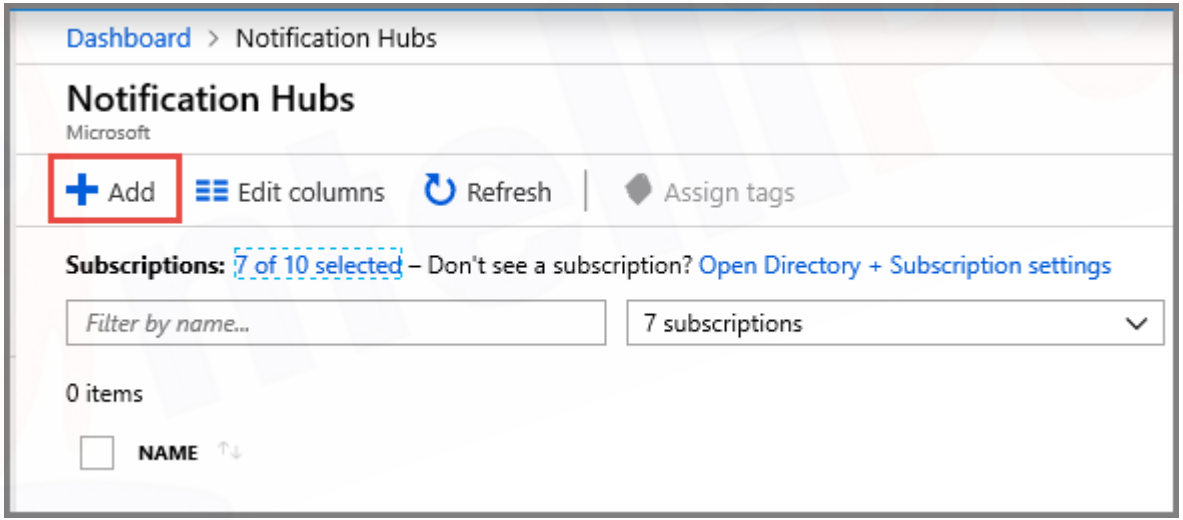
- ★ Sign in to the **Azure portal**.
- ★ Select All services on the left menu, and select Notification Hubs in the Mobile section.
- ★ Select star (*) next to the service name to add it to the FAVORITES section on the left menu.
- ★ After Notification Hubs is added to FAVORITES, select it on the left menu.



Create an Azure notification hub in the Azure portal

Step 1 Continued…

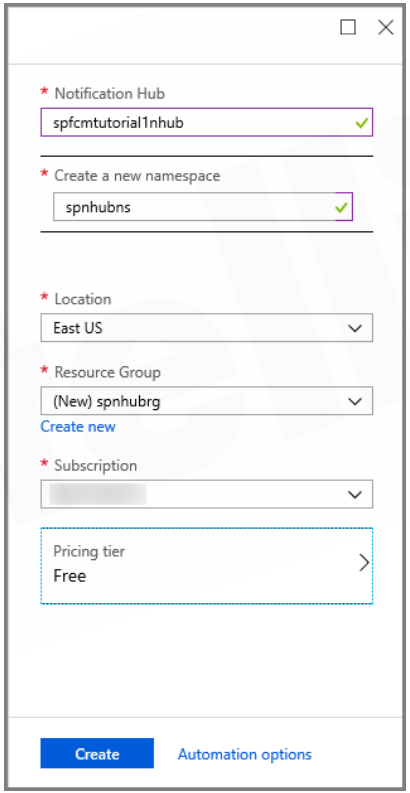
★ On the Notification Hubs page, select Add on the toolbar.



Create an Azure notification hub in the Azure portal

Step 2

- On the Notification Hub page, do the following steps:
- Specify a name for the notification hub.
- Specify a name for the namespace. A namespace contains one or more hubs.

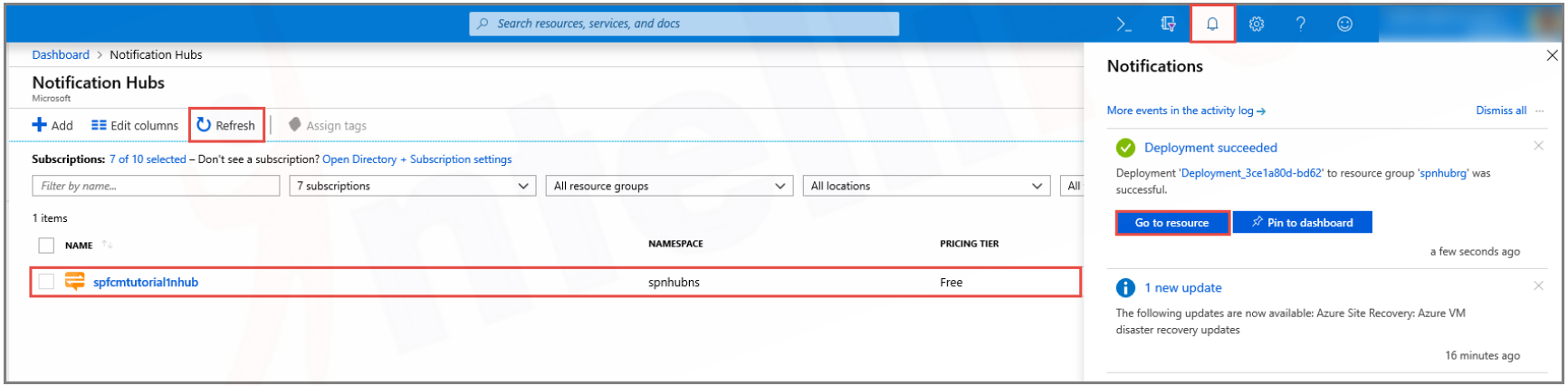


- Select a location in which you want to create the notification hub.
- Select an existing resource group or enter a name for the new resource group.
- Select Create.

Create an Azure notification hub in the Azure portal

Step 2 Continued…

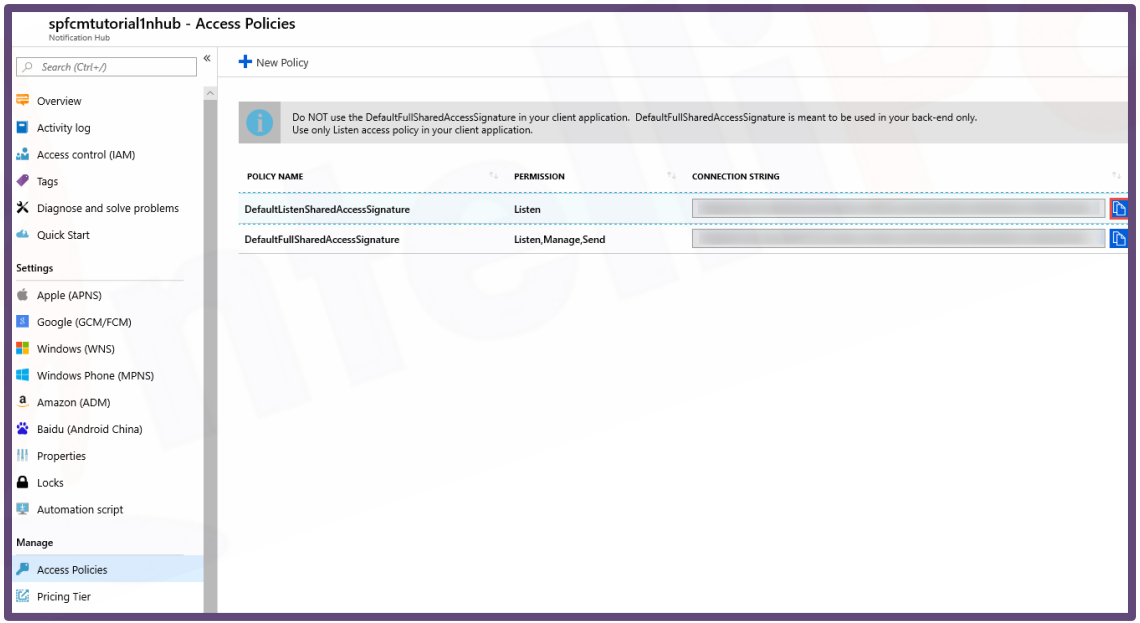
- Select **Notifications** (Bell icon), and select Go to resource. You can also refresh the list in the Notification Hubs page, and select your notification hub.



Create an Azure notification hub in the Azure portal

Step 2 Continued…

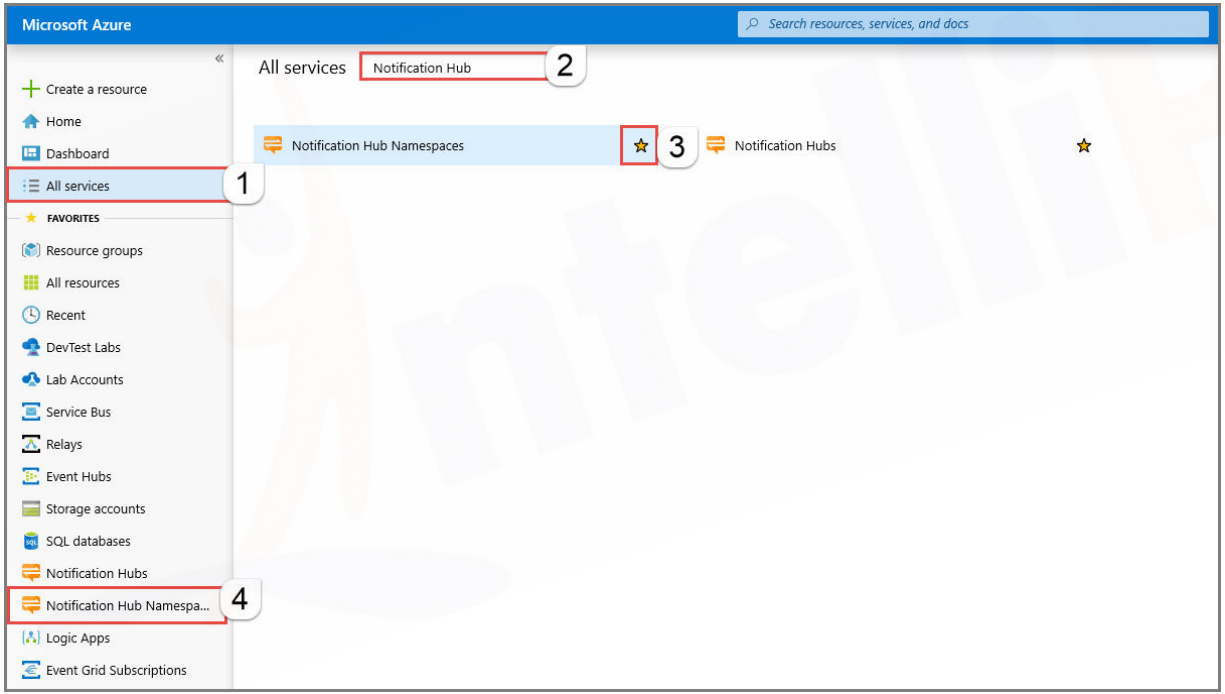
- Select Access Policies from the list. Note the two connection strings that are available to you. You need them to handle push notifications later.



Create an Azure notification hub in the Azure portal

Step 3

★ Create a notification hub in an existing namespace

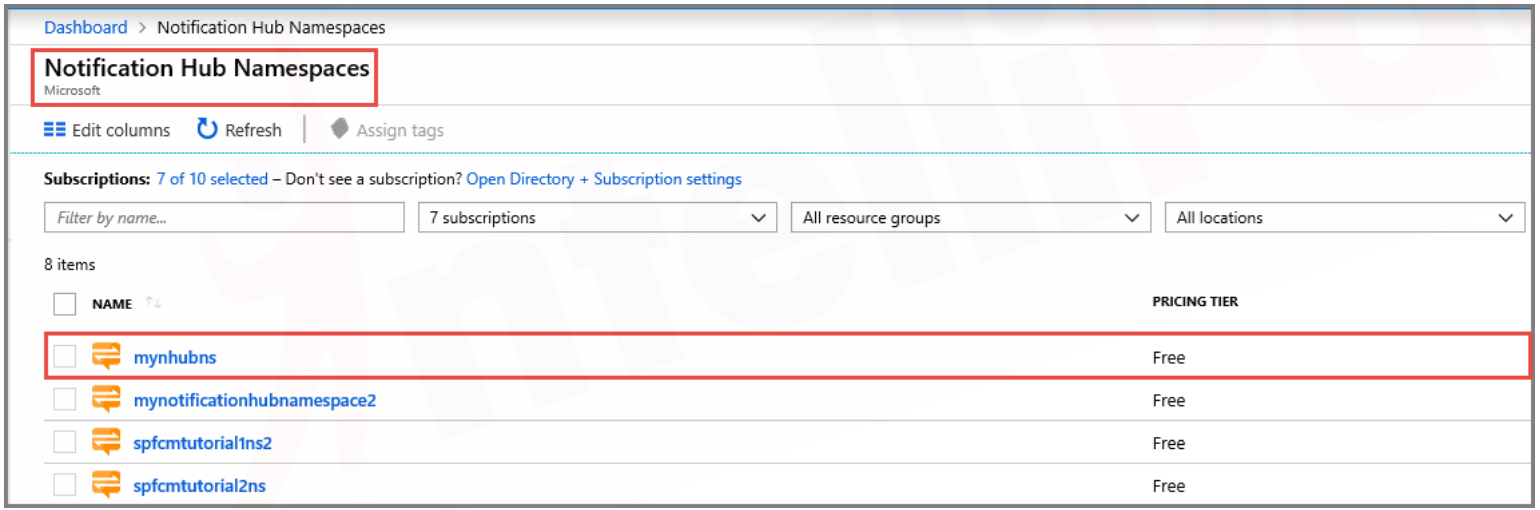


- ★ Sign in to the Azure portal.
- ★ Select All services on the left menu, search for Notification Hub, select star (*) next to Notification Hub Namespaces to add it to the FAVORITES section on the left menu. Select Notification Hub Namespaces.

Create an Azure notification hub in the Azure portal

Step 3

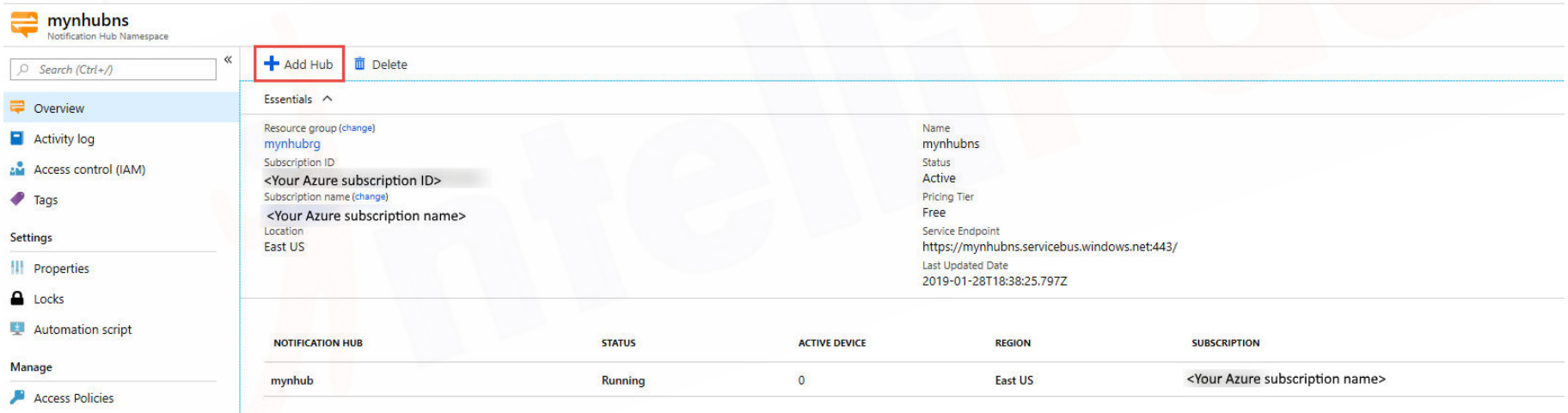
★ On the Notification Hub Namespaces page, select your namespace from the list.



Create an Azure notification hub in the Azure portal

Step 3 Continued…

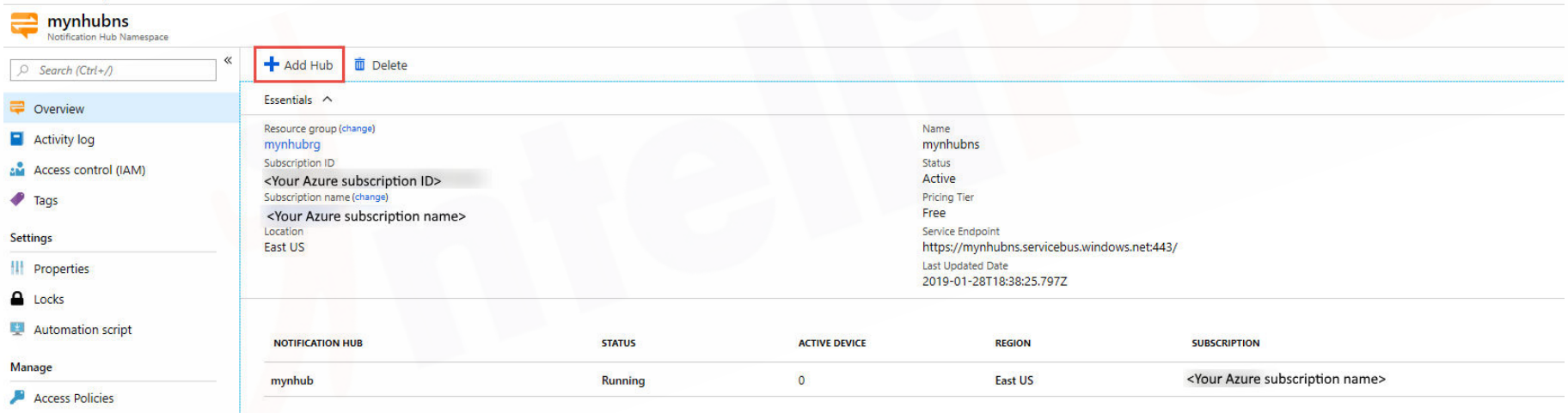
★ On the Notification Hub Namespace page, select Add Hub on the toolbar.



Create an Azure notification hub in the Azure portal

Step 4

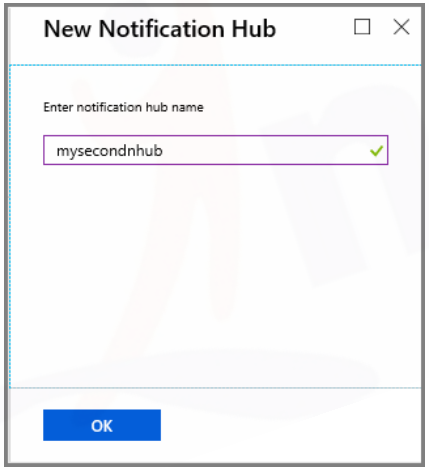
★ On the Notification Hub Namespace page, select Add Hub on the toolbar.

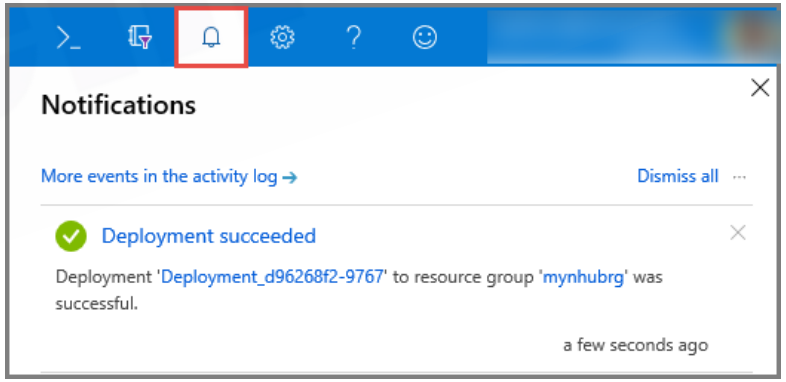


Create an Azure notification hub in the Azure portal

Last Step

- On the New Notification Hub page, enter a name for the notification hub, and select OK.
- Select Notifications (Bell icon) at the top to see the status of the deployment of the new hub. Select X in the right-corner to close the notification window.
- Refresh the Notification Hub Namespaces web page to see your new hub in the list.
- Select your notification hub to see the home page for your notification hub.





India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24/7 Chat with Our Course Advisor