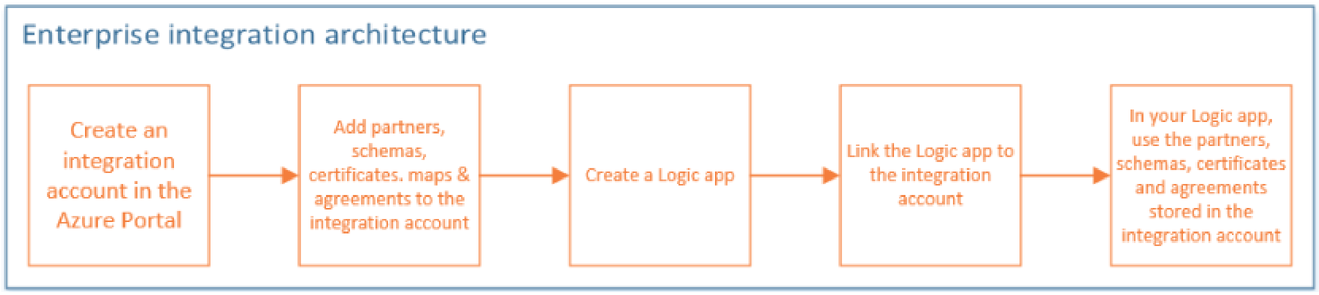
AS2 message processing for B2B enterprise integration in Azure Logic Apps and Integration Account

With Azure supporting business-to-business (B2B) workflows and communication with logic apps, EDI based enterprise integration scenarios can be implemented using Microsoft's cloud based solution, the Enterprise Integration Pack. Variety of standard protocols including EDIFACT, AS2 and X12 are supported. Message security options are also available which include encryption and digital signatures.

You can build and manage B2B apps with the Enterprise Integration Pack through the Logic App Designer in the Azure portal.

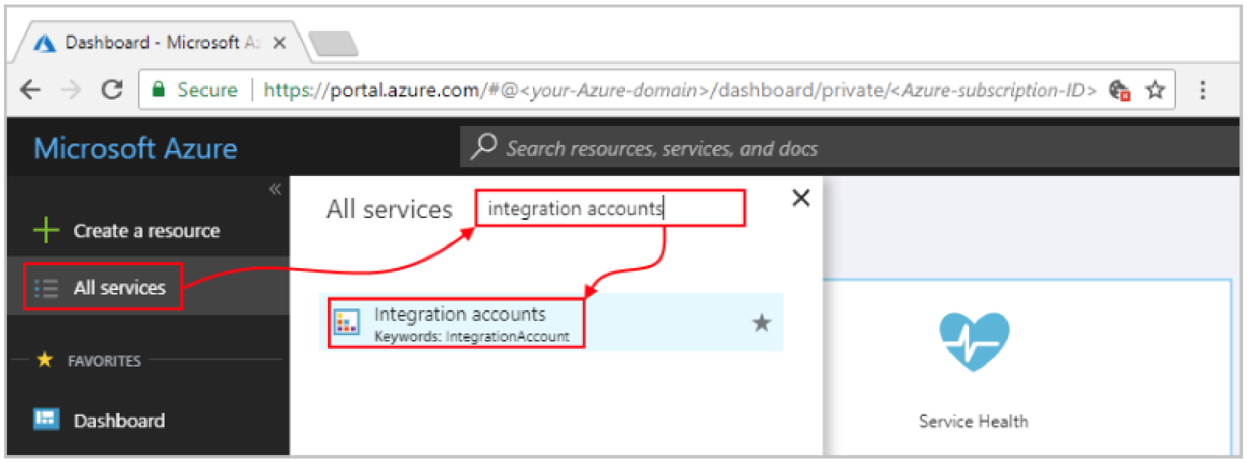
Here are the high-level steps you must take before you can create apps in the Azure portal:


Enterprise integration architecture
Create an integration account in the Azure Portal
Add partners, schemas, certificates. maps & agreements to the integration account
Create a Logic app
Link the Logic app to the integration account
In your Logic app, use the partners, schemas, certificates and agreements stored in the integration account

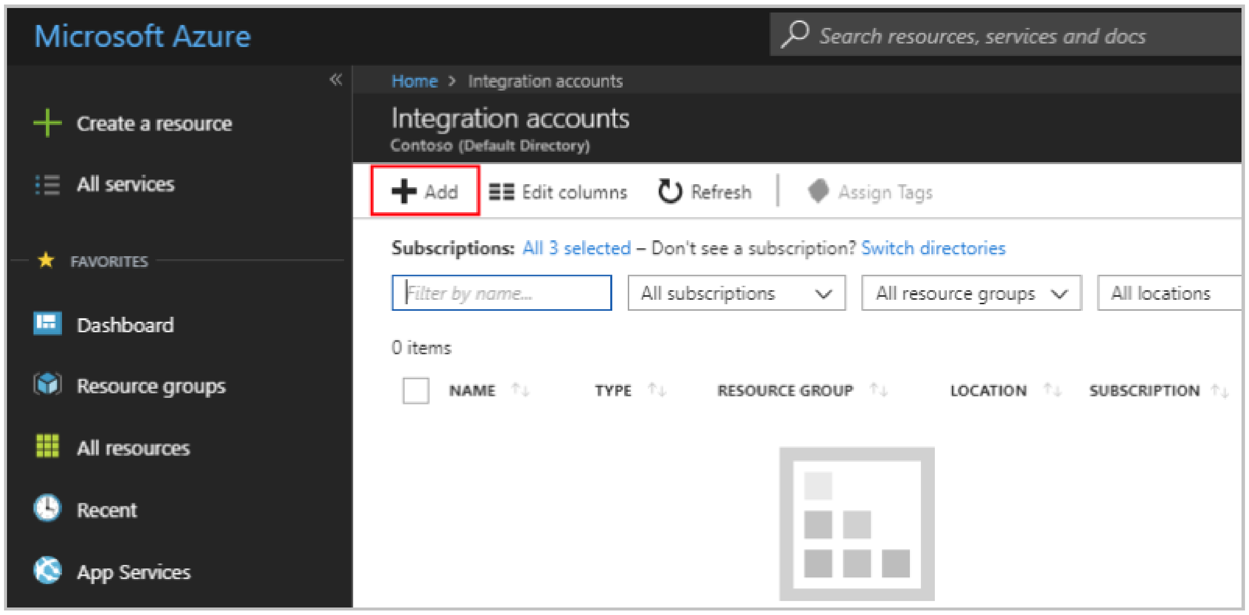
This blog demonstrates AS2 message processing using logic apps. It includes creation of Integration Account, creation of partners and agreements and creation of logic apps to send and receive AS2 messages from one partner to the other.

Create Integration Account-

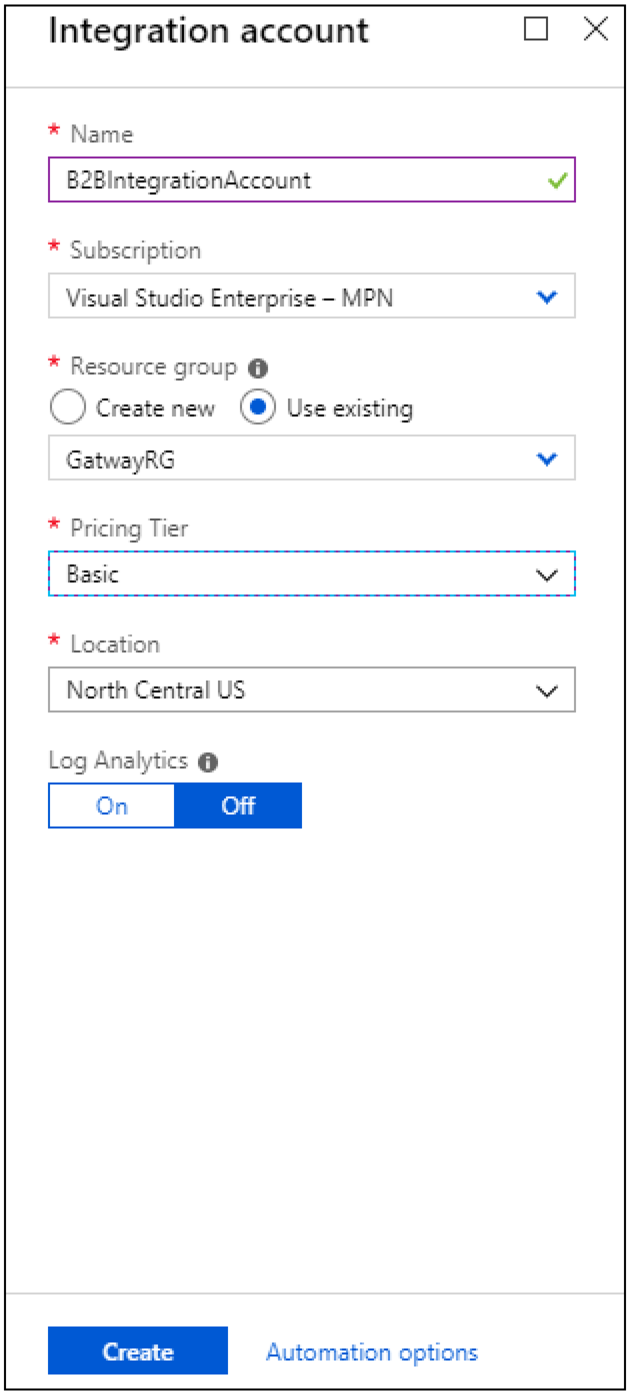
1. From the main Azure menu, select **All services**. In the search box, enter "integration accounts" as your filter, and select **Integration accounts**.



2. Under **Integration accounts**, choose **Add**.



3. Provide information about your integration account and choose Create:


Integration account

* Name

B2BIntegrationAccount

* Subscription

Visual Studio Enterprise – MPN

* Resource group

Create new [✓ Use existing]

GatwayRG

* Pricing Tier

Basic

* Location

North Central US

Log Analytics

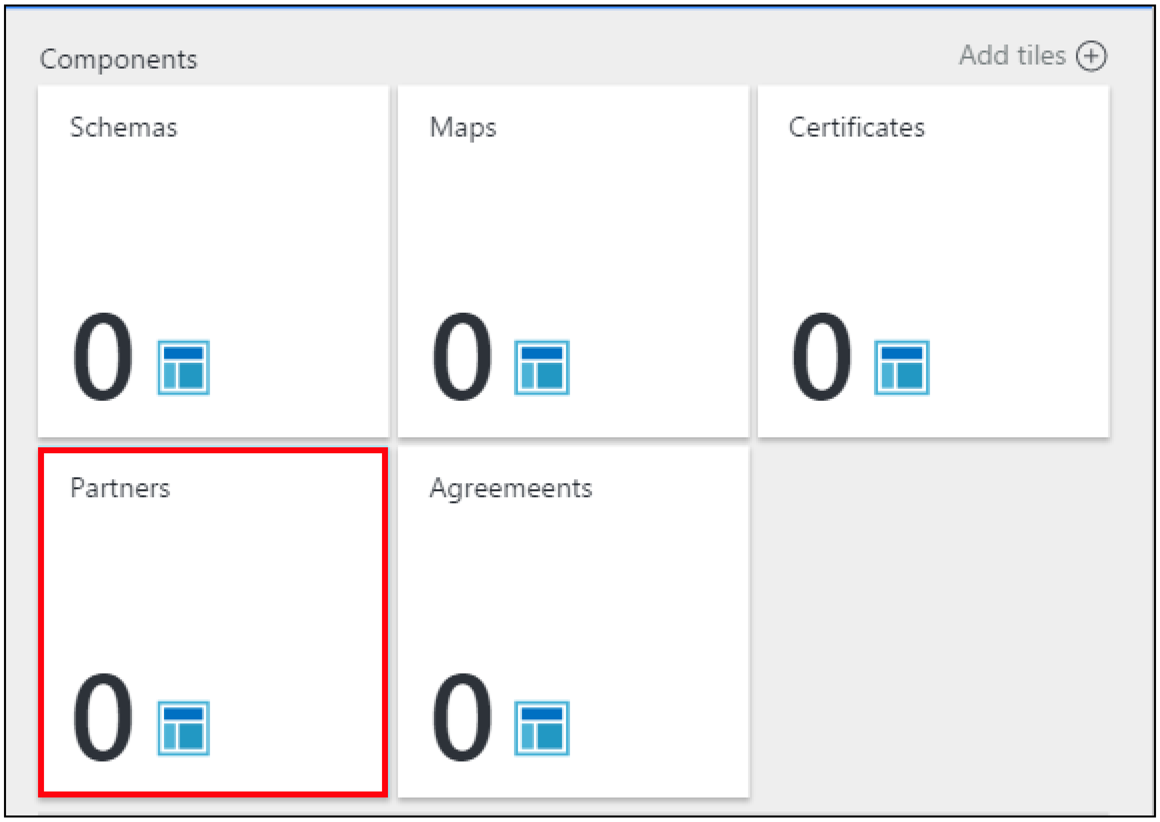
On [✓ Off]

Note: You can also choose Free Pricing Tier for this solution.

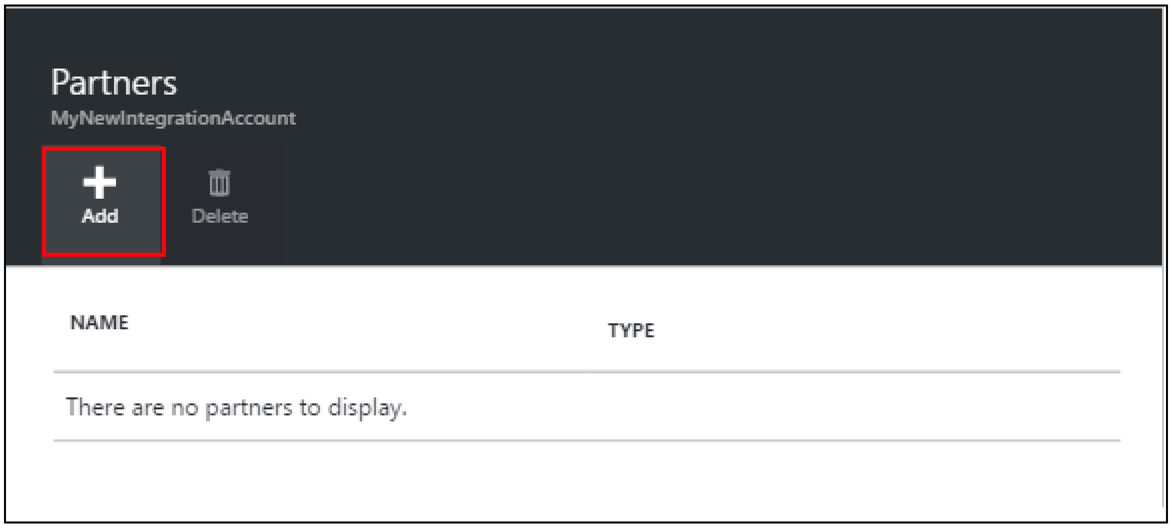
After Azure deploys your integration account to the selected location, which usually finishes within one minute, Azure opens your integration account.

Create Partner

1. In Integration Account, Choose the **Partners** tile.

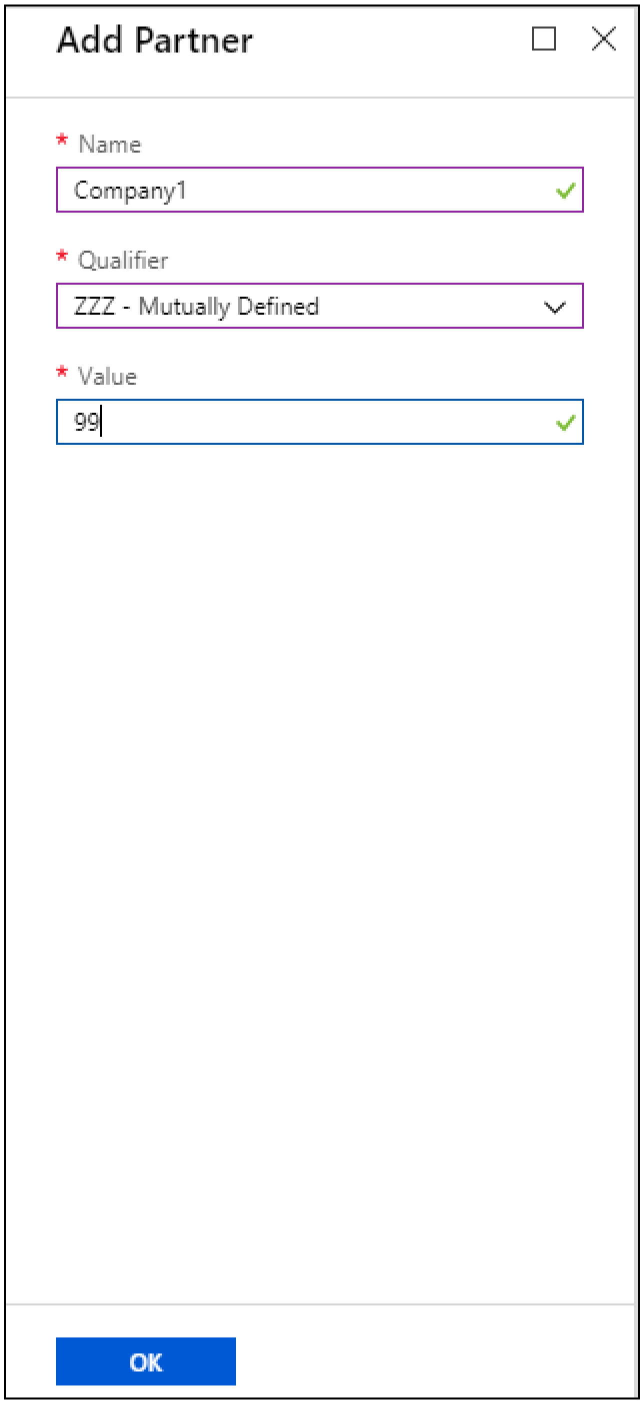


2. Under Partners, choose Add.



NAME	TYPE
There are no partners to display.	

3. Enter a name for your partner, then select a Qualifier. Enter a Value to identify documents that your apps receive. When you're done, choose OK.


Add Partner

* Name

Company1

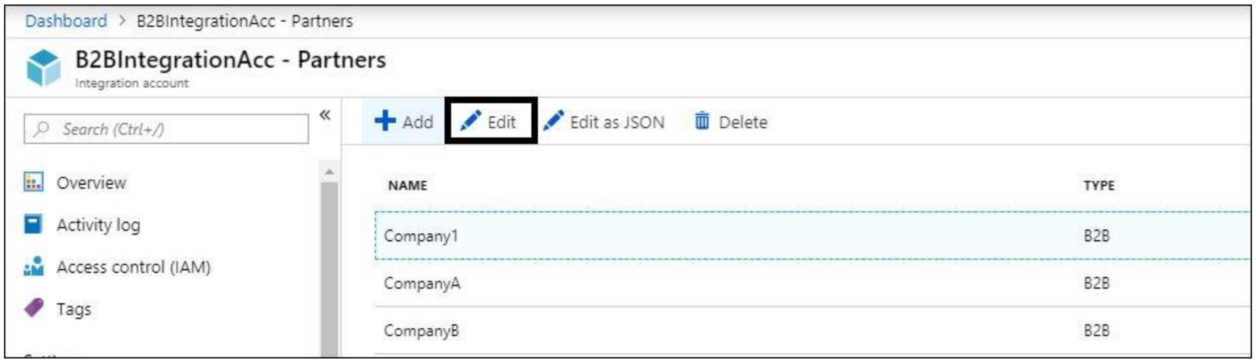
* Qualifier

ZZZ - Mutually Defined

* Value

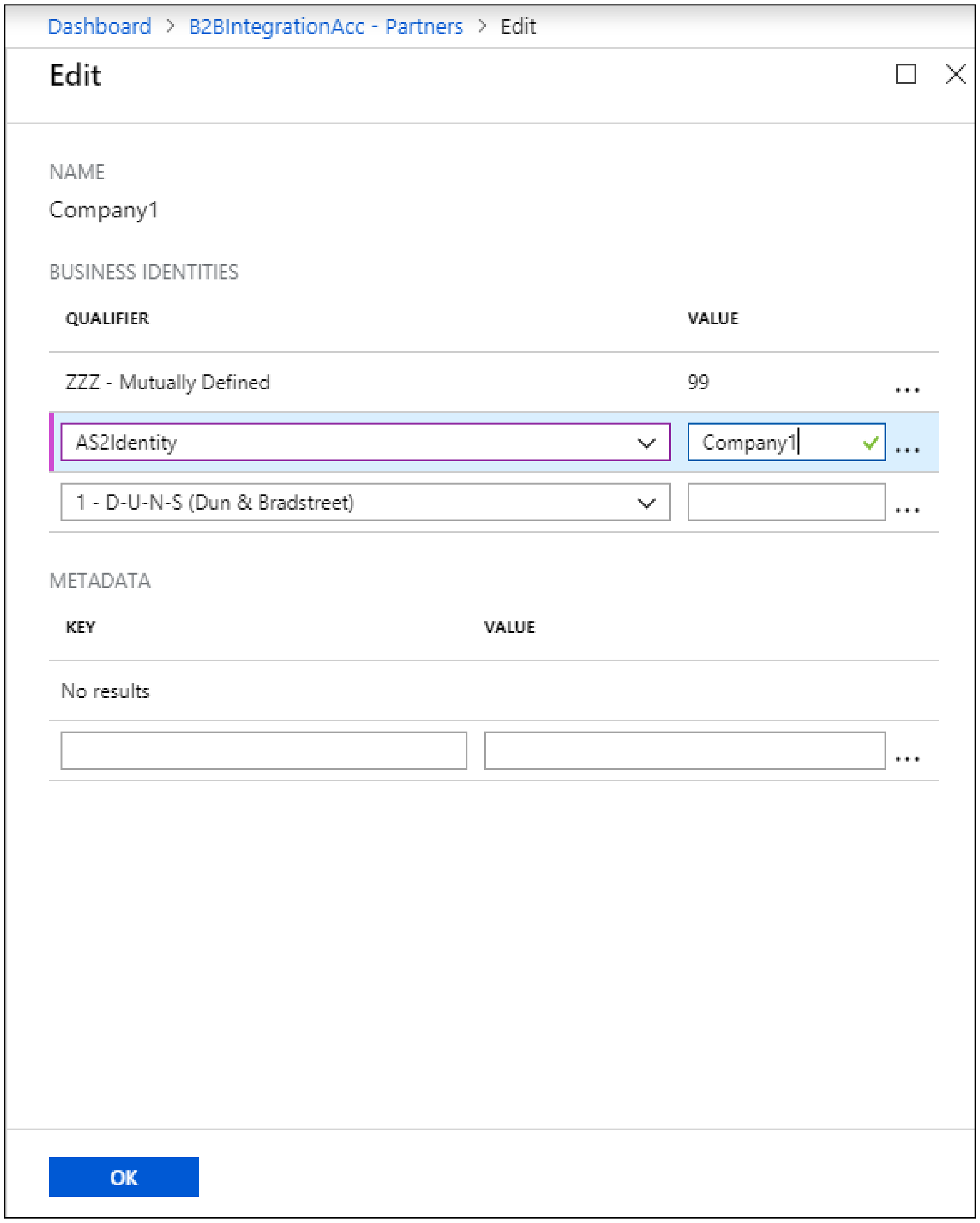
99

If you want to add more than 1 qualifier, click OK. Select the created partner and click Edit.



NAME	TYPE
Company1	B2B
CompanyA	B2B
CompanyB	B2B

4. Add the new qualifier & value and click ok.



NAME
Company1

BUSINESS IDENTITIES

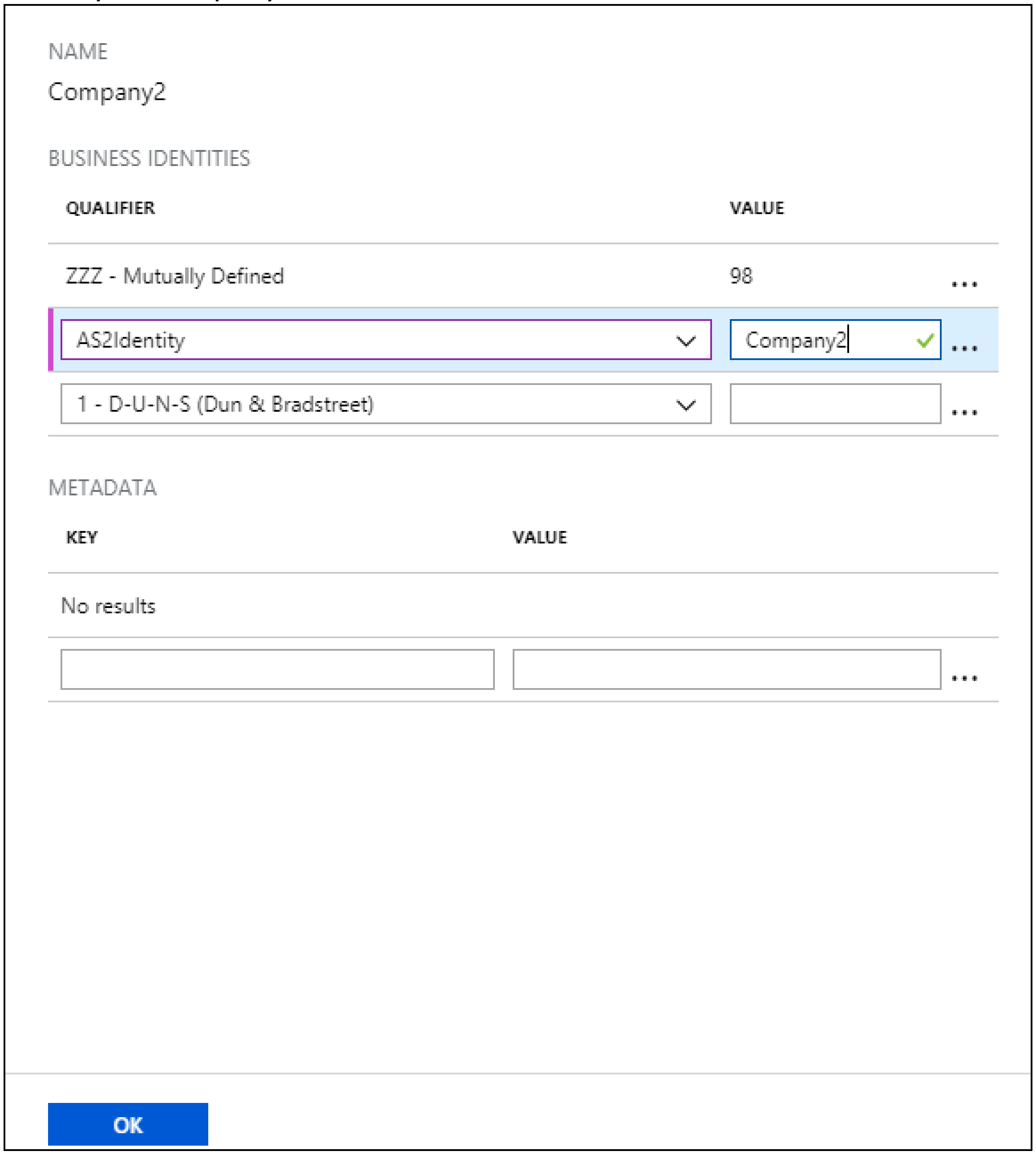
QUALIFIER	VALUE
ZZZ - Mutually Defined	99
AS2Identity	Company1
1 - D-U-N-S (Dun & Bradstreet)	

METADATA

KEY	VALUE
No results	

Here we will add AS2Identity Qualifier with Partner name as value (Company1 in this case). This information will be used for resolving agreement and message processing.

5. Similarly add one more partner Company2 with ZZ qualifier as 98 and AS2 identity as Company2.



NAME

Company2

BUSINESS IDENTITIES

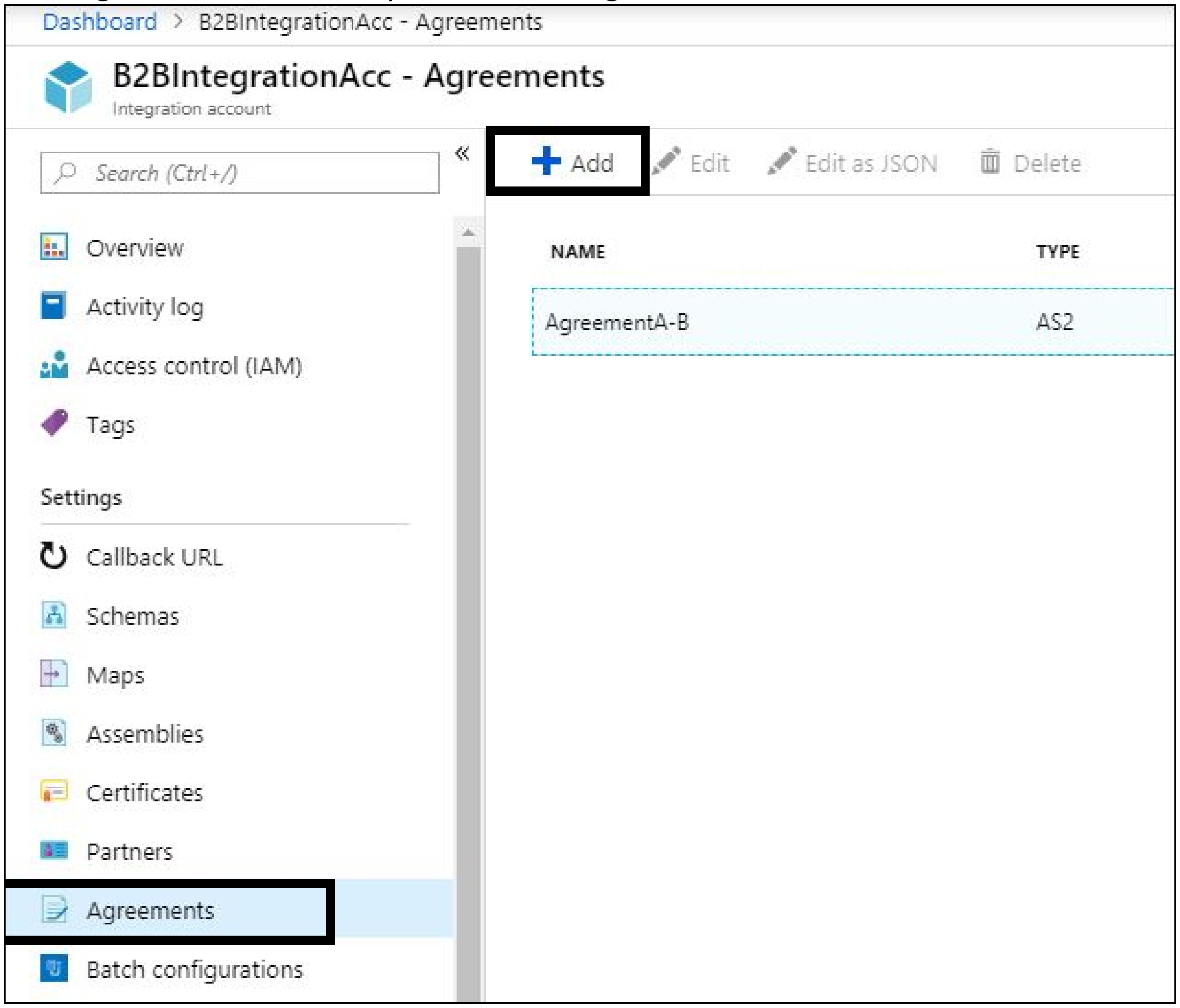
QUALIFIER	VALUE
ZZZ - Mutually Defined	98
AS2Identity	Company2
1 - D-U-N-S (Dun & Bradstreet)	

METADATA

KEY	VALUE
No results	

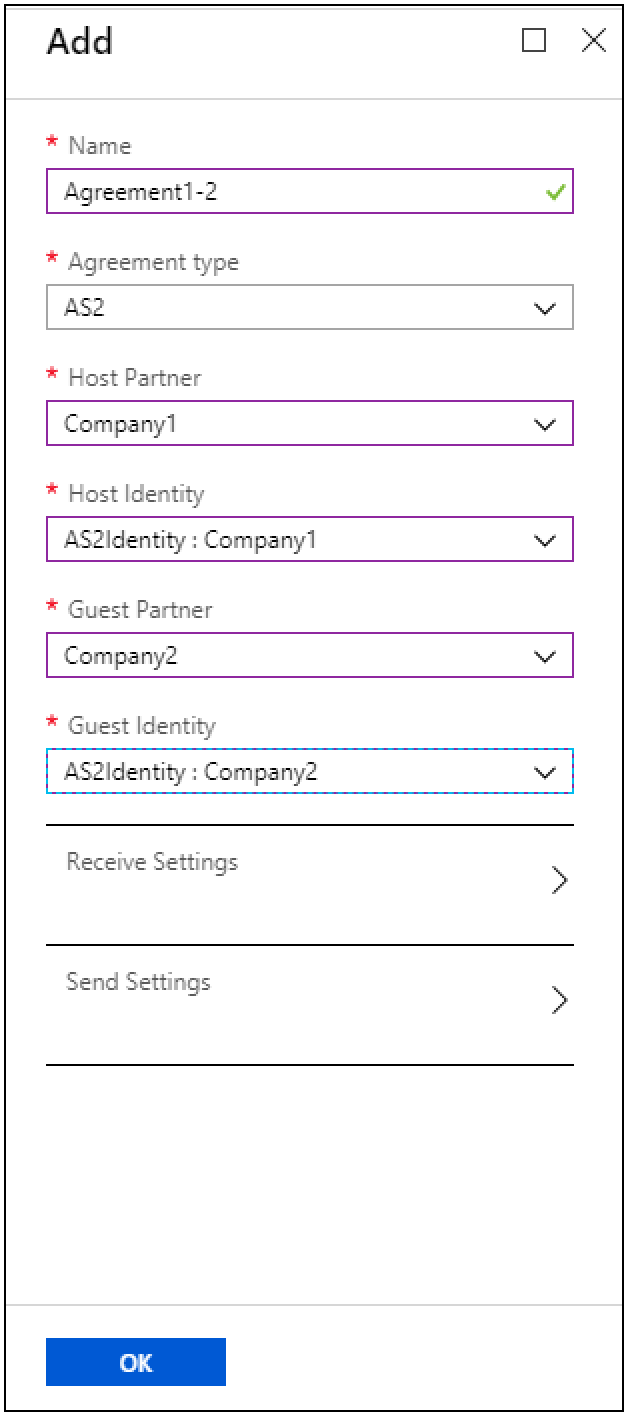
Create Agreement-

1. In Integration Account left pane, choose Agreements and click Add.



NAME	TYPE
AgreementA-B	AS2

2. Under **Add**, enter a **Name** for your agreement. For **Agreement type**, select **AS2**. Select the **Host Partner**, **Host Identity**, **Guest Partner**, and **Guest Identity** for your agreement.


Add

* Name

Agreement1-2

* Agreement type

AS2

* Host Partner

Company1

* Host Identity

AS2Identity : Company1

* Guest Partner

Company2

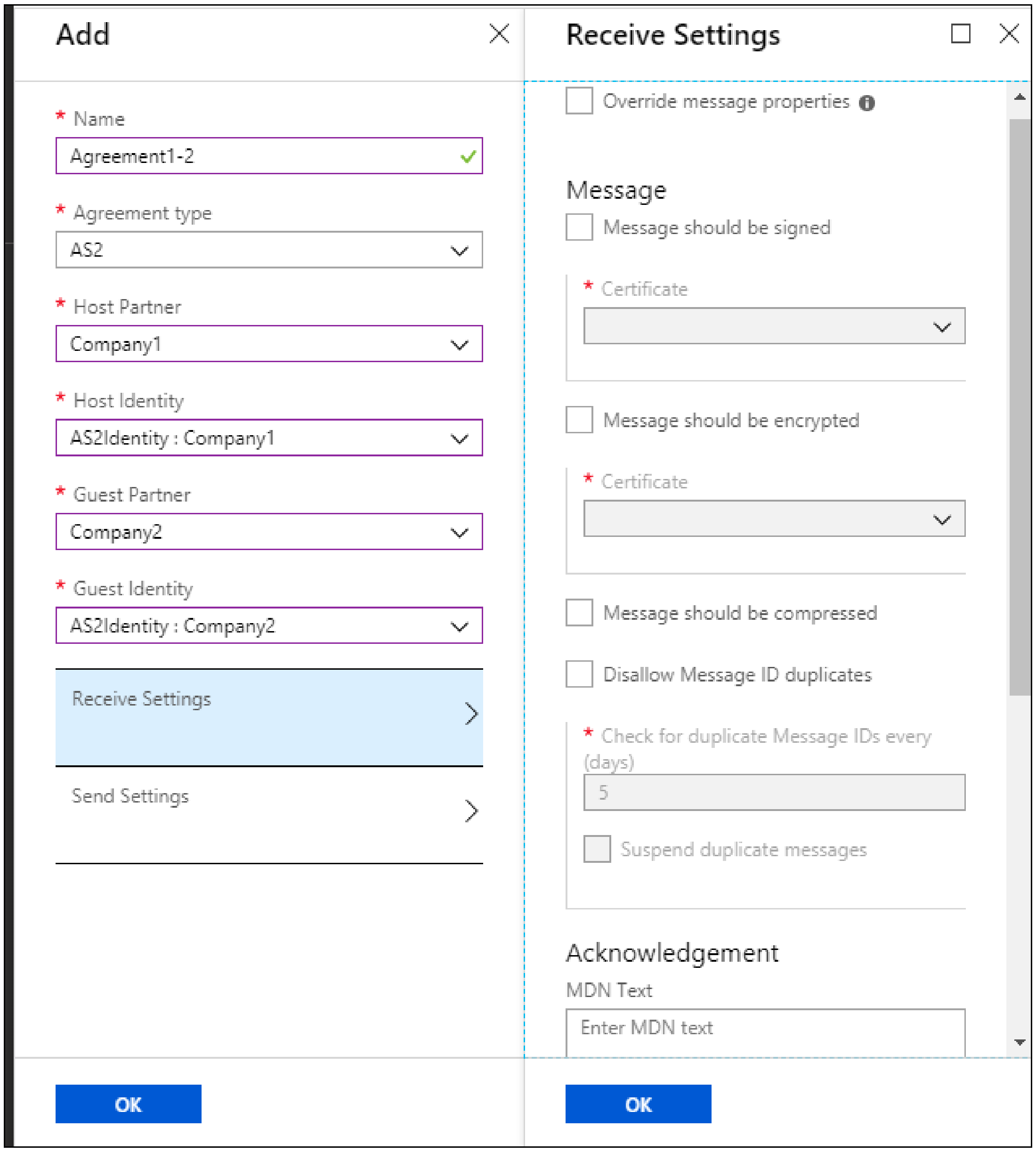
* Guest Identity

AS2Identity : Company2

Property	**Description**
Name	Name of the agreement
Agreement Type	Should be AS2
Host Partner	An agreement needs both a host and guest partner. The host partner represents the organization that configures the agreement.
Host Identity	An identifier for the host partner
Guest Partner	An agreement needs both a host and guest partner. The guest partner represents the organization that's doing business with the host partner.
Guest Identity	An identifier for the guest partner
Receive Settings	These properties apply to all messages received by an agreement.
Send Settings	These properties apply to all messages sent by an agreement.

3. You can configure how this agreement identifies and handles incoming messages received from your partner through this agreement.

   Under Add, select Receive Settings. Configure these properties based on your agreement with the partner that exchanges messages with you.



Add

* Name
Agreement1-2

* Agreement type
AS2

* Host Partner
Company1

* Host Identity
AS2Identity : Company1

* Guest Partner
Company2

* Guest Identity
AS2Identity : Company2

Receive Settings

Send Settings

OK

Receive Settings

- [ ] Override message properties

Message

- [ ] Message should be signed

* Certificate

- [ ] Message should be encrypted

* Certificate

- [ ] Message should be compressed
- [ ] Disallow Message ID duplicates

* Check for duplicate Message IDs every (days)
5

- [ ] Suspend duplicate messages

Acknowledgement

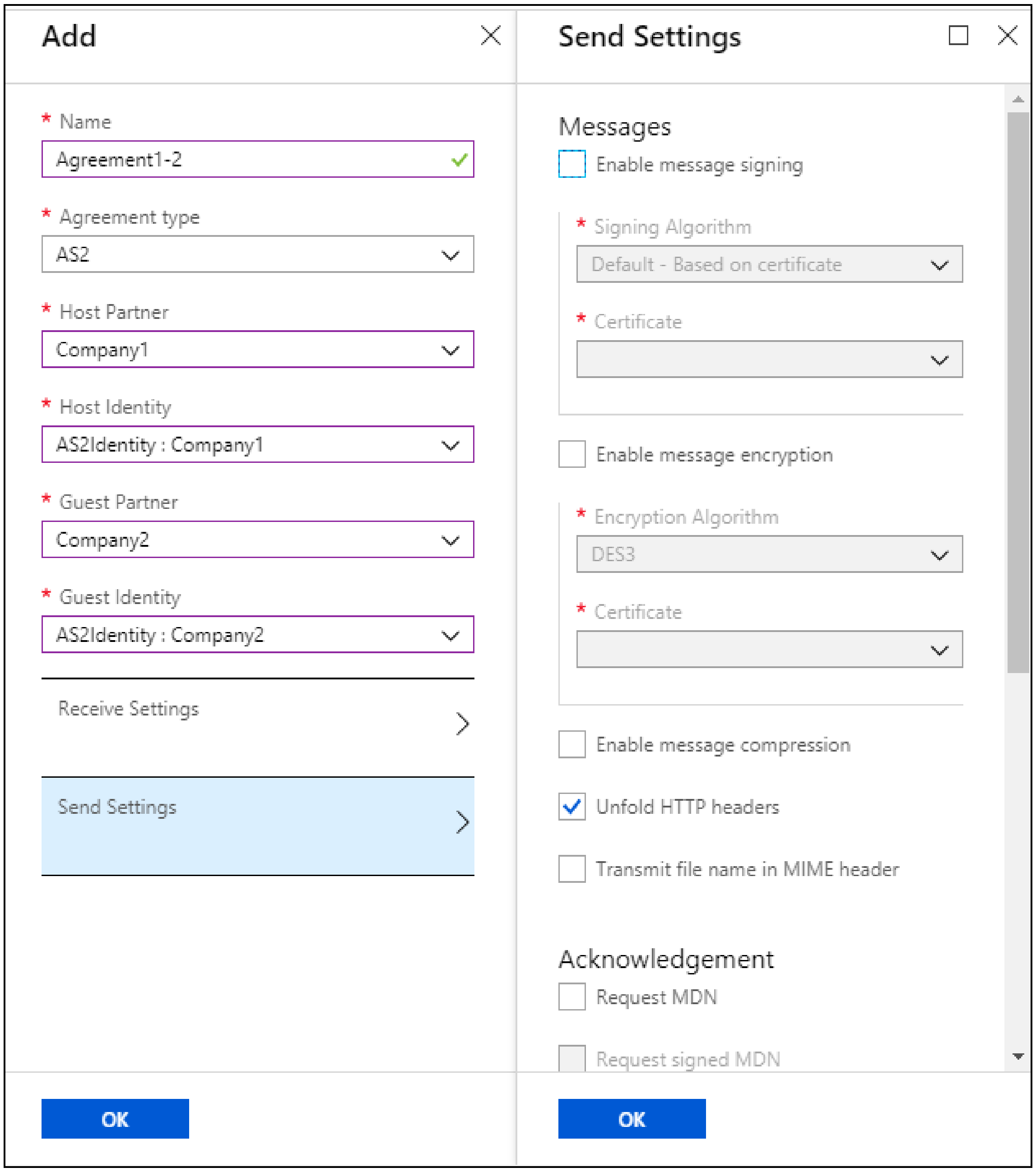
MDN Text
Enter MDN text

OK

We will use the default settings for this PoC.

4. Similarly, you can configure how this agreement identifies and handles outgoing messages that you send to your partners through this agreement.

   Under Add, select Send Settings. Configure these properties based on your agreement with the partner that exchanges messages with you.


Add

* Name
Agreement1-2

* Agreement type
AS2

* Host Partner
Company1

* Host Identity
AS2Identity : Company1

* Guest Partner
Company2

* Guest Identity
AS2Identity : Company2

Receive Settings

Send Settings

OK

Send Settings

Messages
☐ Enable message signing

* Signing Algorithm
Default - Based on certificate

* Certificate

☐ Enable message encryption

* Encryption Algorithm
DES3

* Certificate

☐ Enable message compression

✓ Unfold HTTP headers

☐ Transmit file name in MIME header

Acknowledgement
☐ Request MDN

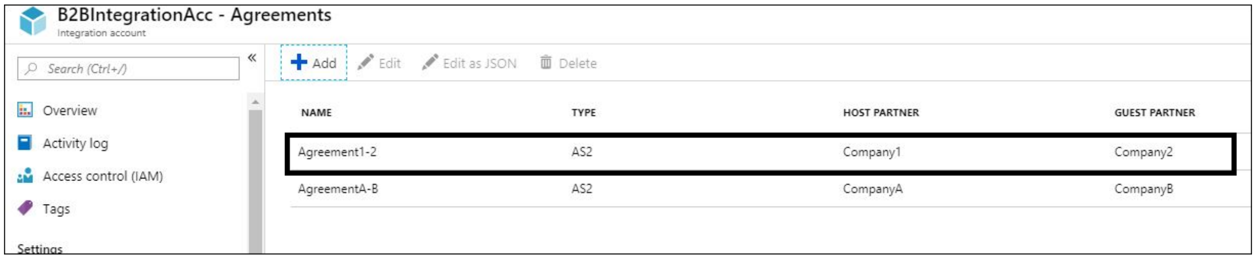
☐ Request signed MDN

OK

We will use the default settings for this PoC.

5. After you're done, make sure to save your settings by choosing OK. After you finish setting all your agreement properties, on the Add page, choose OK to finish creating your agreement and return to your integration account.

Your newly added agreement now appears in your **Agreements** list.



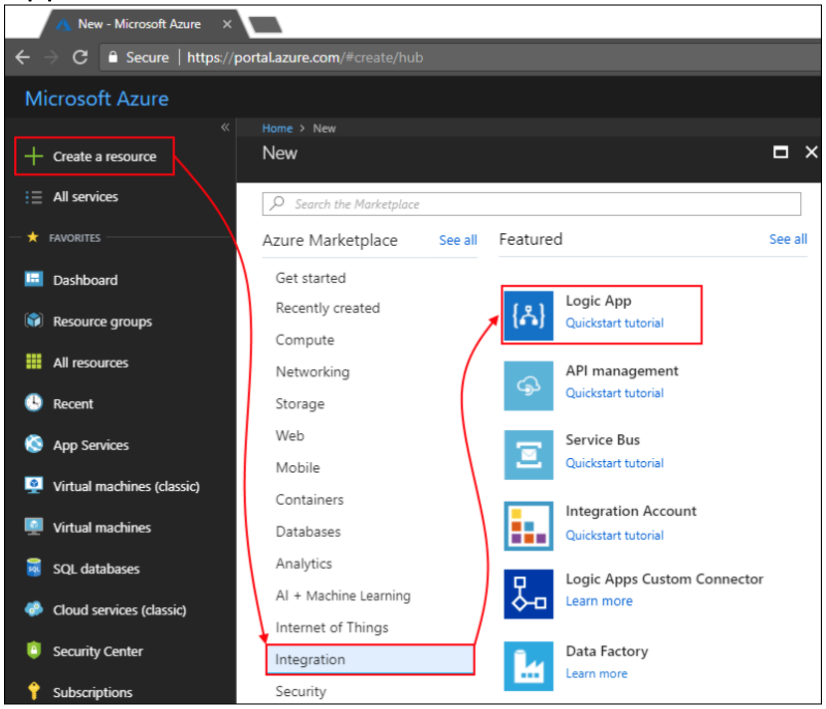
NAME	TYPE	HOST PARTNER	GUEST PARTNER
Agreement1-2	AS2	Company1	Company2
AgreementA-B	AS2	CompanyA	CompanyB

Receive B2B data with Azure Logic Apps and Enterprise Integration Pack-

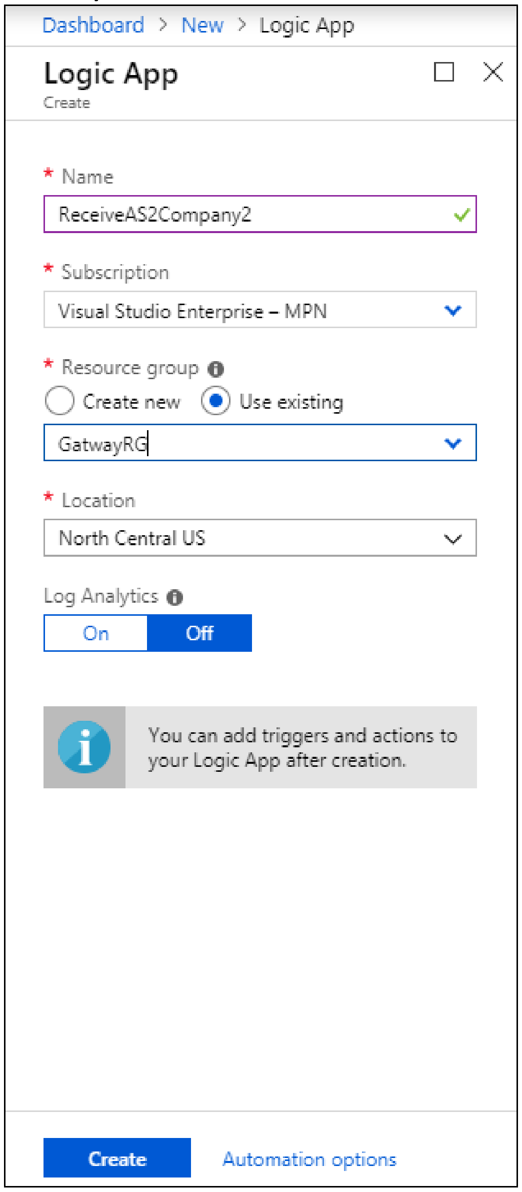
After you create an integration account that has partners and agreements, you are ready to create a business to business (B2B) workflow for your logic app with the Enterprise Integration Pack.
We will create 2 logic apps, one to send the AS2 messages and the other to receive the AS2 messages which are sent by 1st Logic App.

Create Logic App to Receive AS2 messages-

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



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



Logic App

Create

* Name

ReceiveAS2Company2

* Subscription

Visual Studio Enterprise – MPN

* Resource group

Create new [✓ Use existing]

GatwayRG

* Location

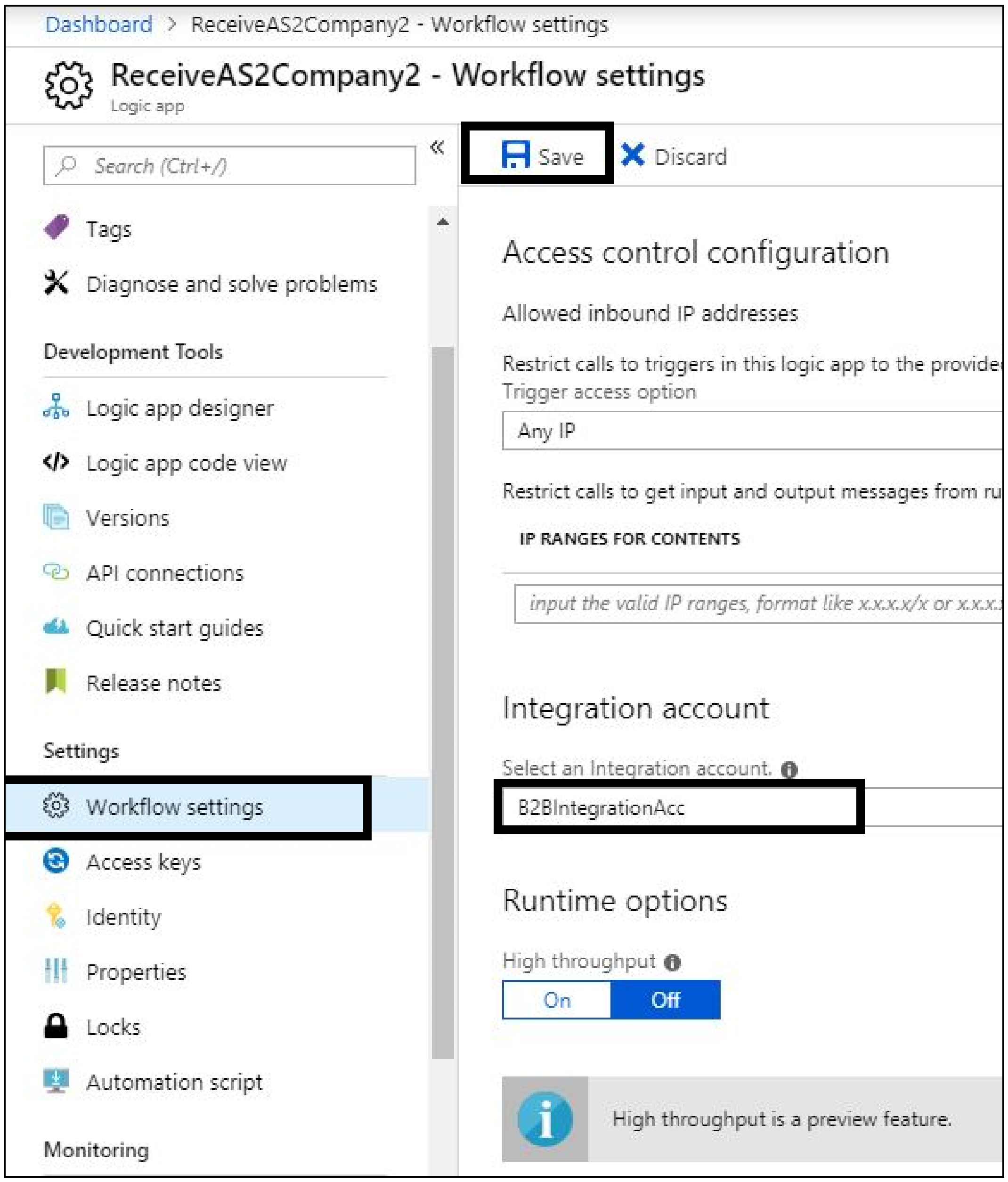
North Central US

Log Analytics

On [✓ Off]

You can add triggers and actions to your Logic App after creation.

3. After Azure deploys your app, link it to your Integration account. Under Workflow settings, choose your Integration Account and click Save.


Dashboard > ReceiveAS2Company2 - Workflow settings

ReceiveAS2Company2 - Workflow settings
Logic app

Access control configuration

Allowed inbound IP addresses

Restrict calls to triggers in this logic app to the provide
Trigger access option: Any IP

Restrict calls to get input and output messages from ru

IP RANGES FOR CONTENTS

input the valid IP ranges, format like x.x.x.x/x or x.x.x.

Integration account

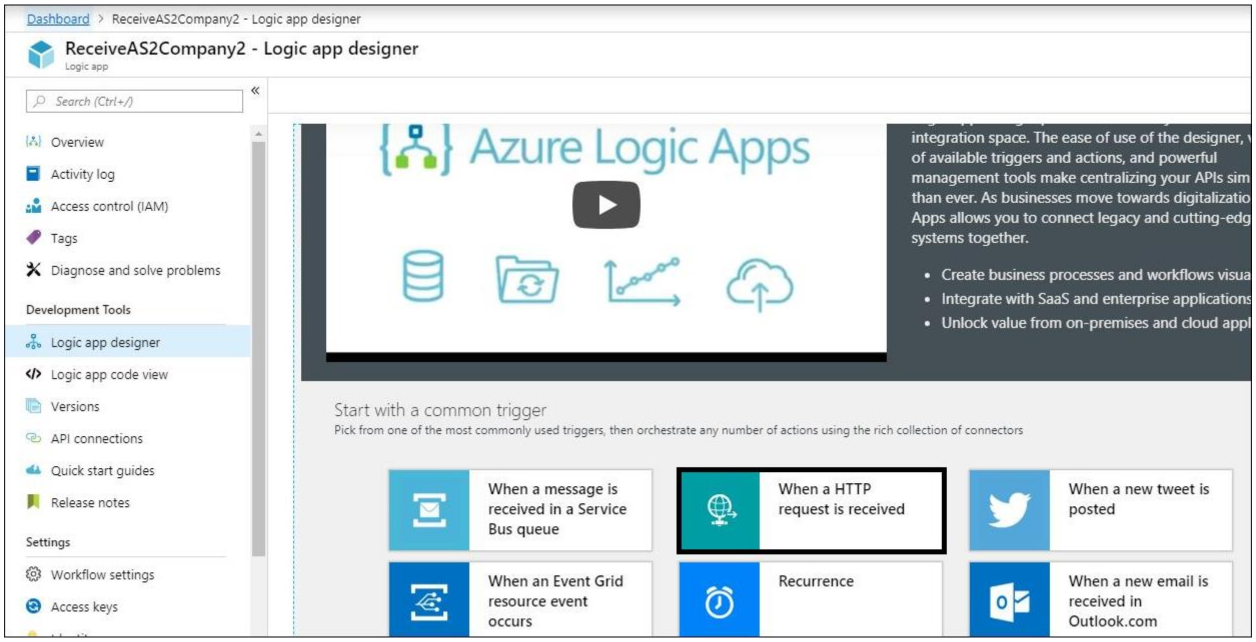
Select an Integration account.: B2BIntegrationAcc

Runtime options

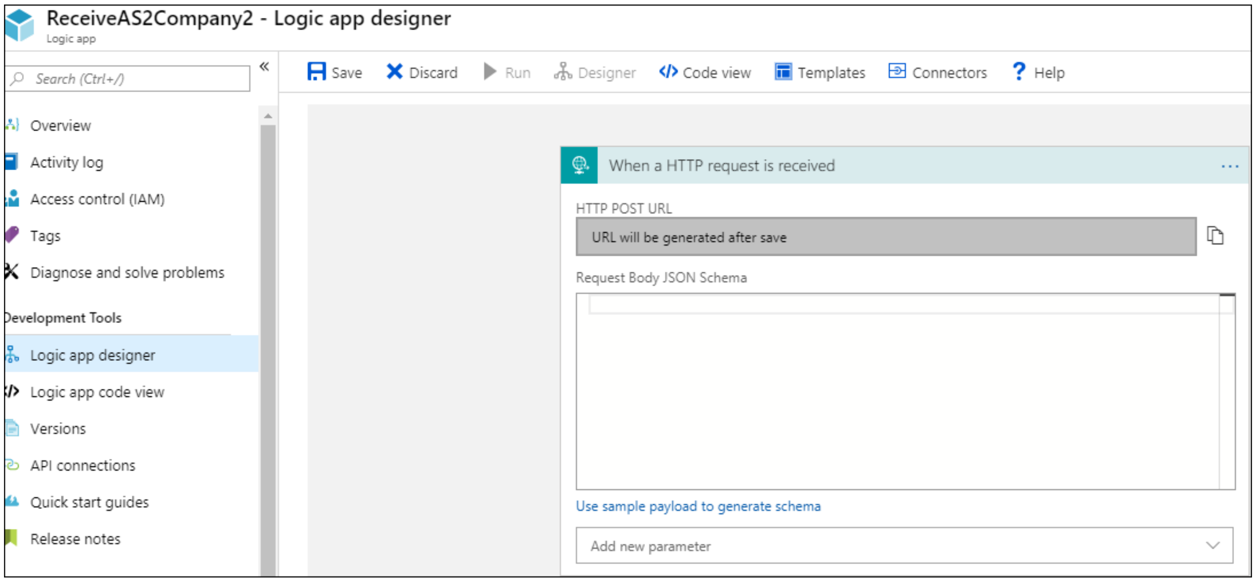
High throughput: On [✓ Off]

High throughput is a preview feature.

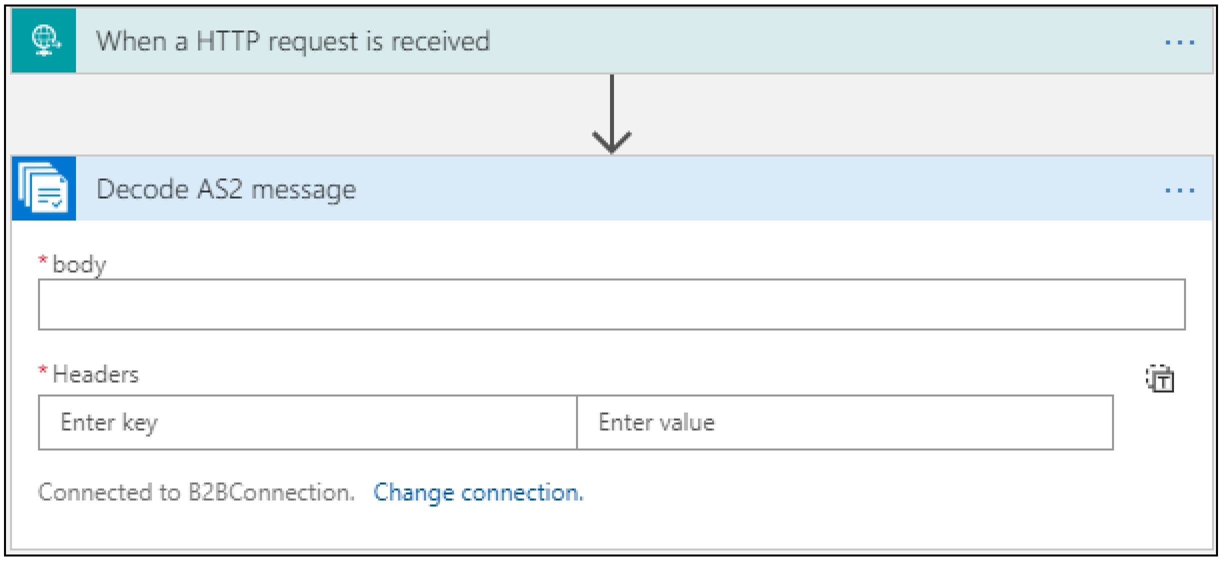
4. Open Logic Apps Designer. Choose the Trigger- **When a HTTP request is received**.


Dashboard > ReceiveAS2Company2 - Logic app designer
ReceiveAS2Company2 - Logic app designer
Logic app
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Development Tools
Logic app designer
Logic app code view
Versions
API connections
Quick start guides
Release notes
Settings
Workflow settings
Access keys
Azure Logic Apps
integration space. The ease of use of the designer,
of available triggers and actions, and powerful
management tools make centralizing your APIs sim
than ever. As businesses move towards digitalizatio
Apps allows you to connect legacy and cutting-edg
systems together.
Create business processes and workflows visua
Integrate with SaaS and enterprise applications
Unlock value from on-premises and cloud appl
Start with a common trigger
Pick from one of the most commonly used triggers, then orchestrate any number of actions using the rich collection of connectors
When a message is received in a Service Bus queue
When a HTTP request is received
When a new tweet is posted
When an Event Grid resource event occurs
Recurrence
When a new email is received in Outlook.com

5. In the logic app designer, leave the Request Body JSON Schema blank. Click Save to generate the HTTP Post URL which will be used to receive the AS2 messages requests. Make a note of this URL to use later.

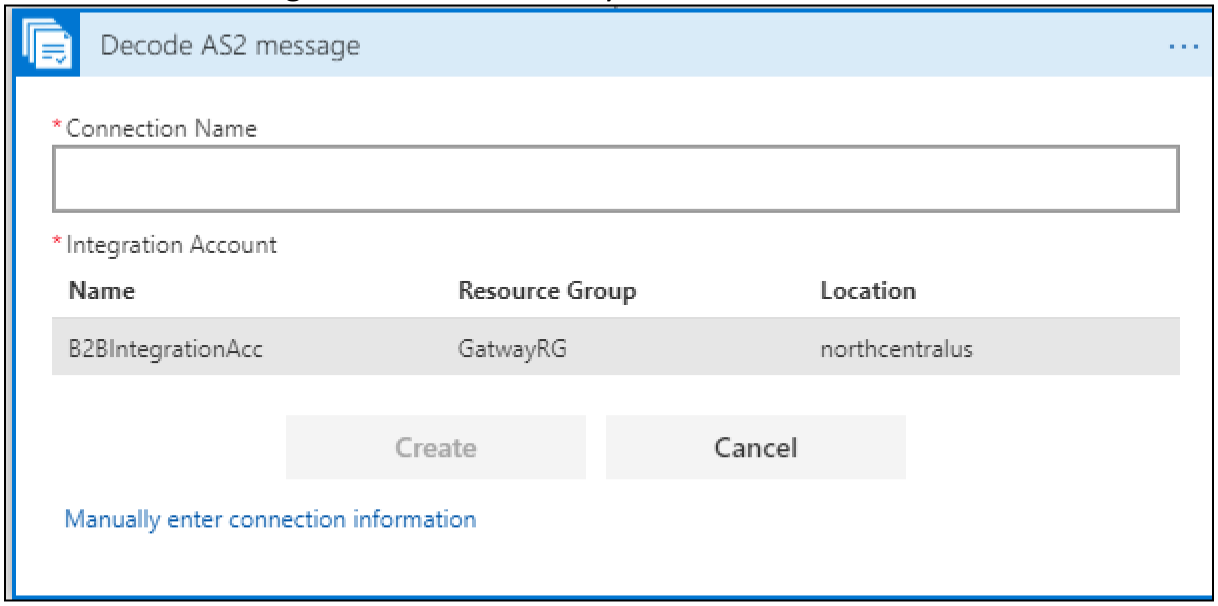


6. In the search box, enter "AS2" for your filter. Select AS2 - Decode AS2 message.



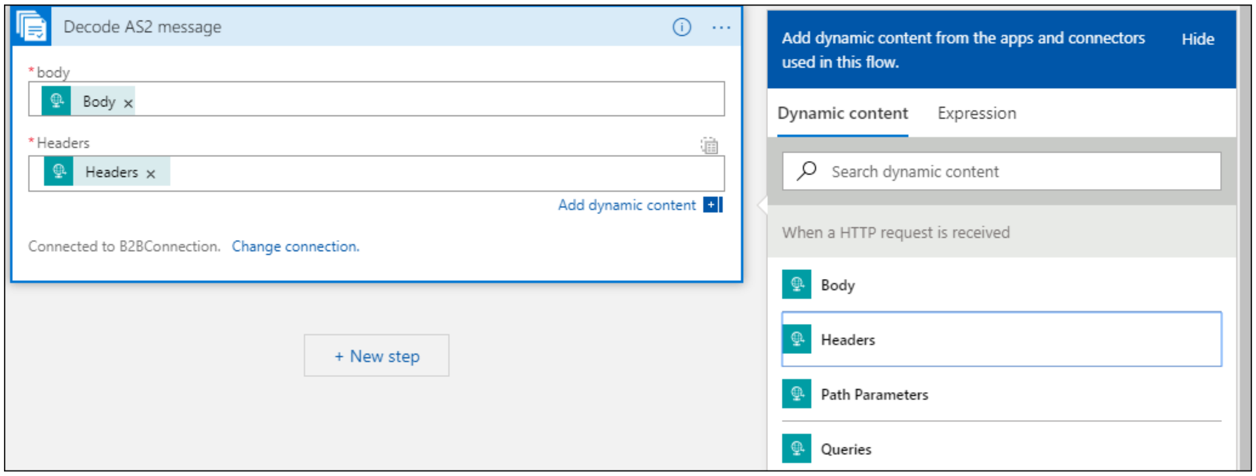
Connected to B2BConnection. Change connection.

7. If you didn't previously create any connections to your integration account, you're prompted to create that connection now. Name your connection and select the integration account that you want to connect.

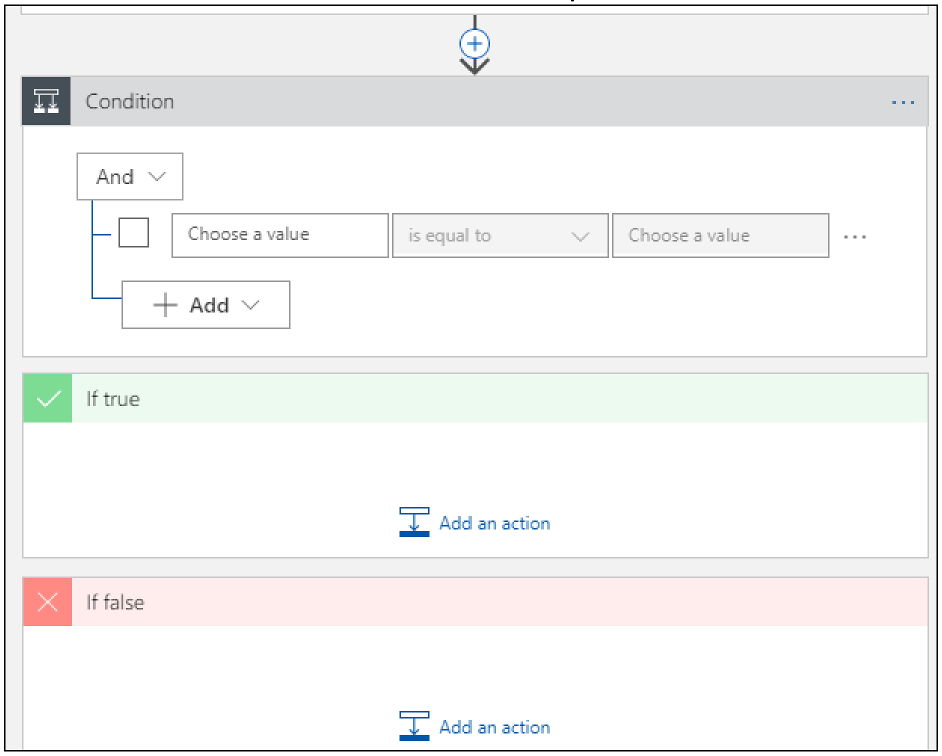


Name	Resource Group	Location
B2BIntegrationAcc	GatwayRG	northcentralus

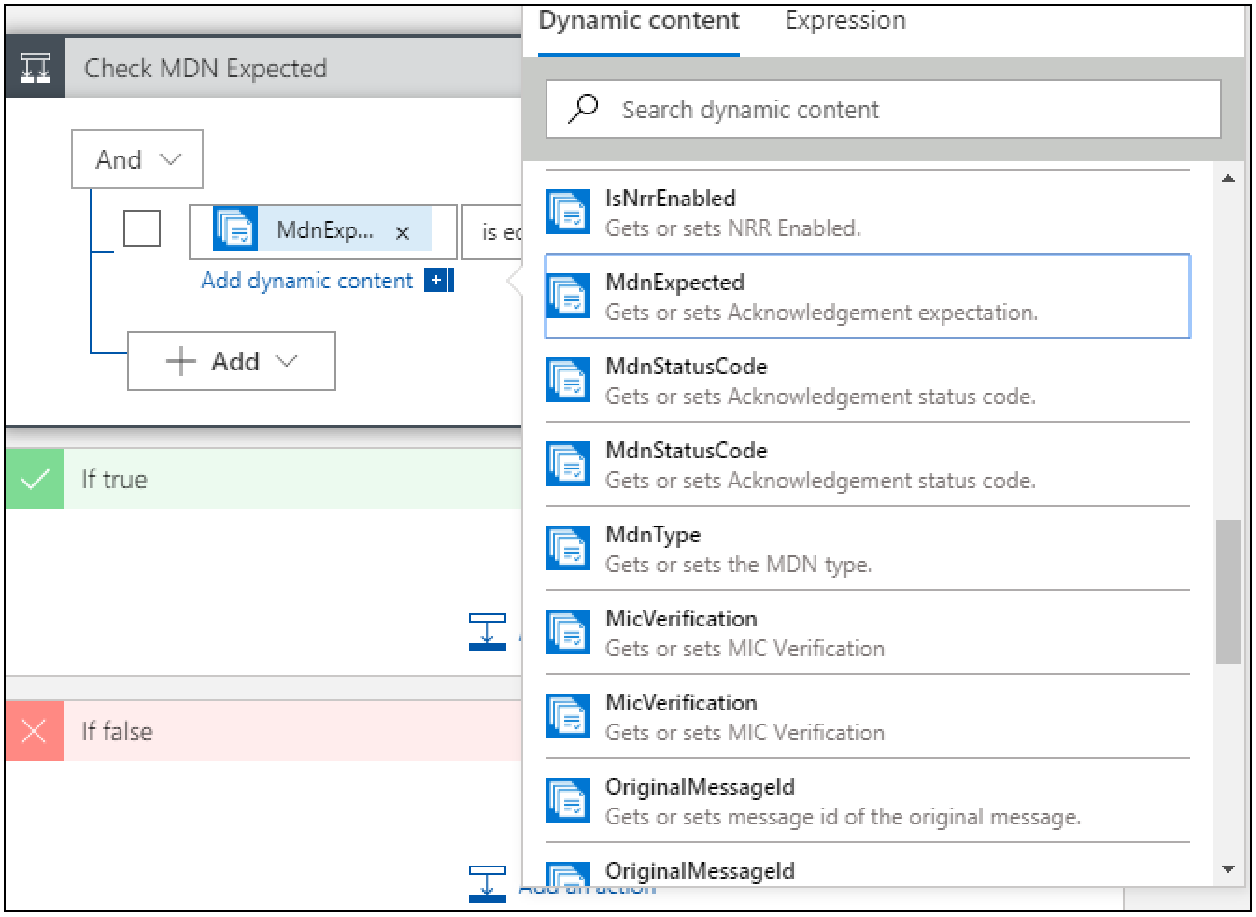
8. After your connection is created, as shown in this example, select Body and Headers from the Request outputs.



9. Add Condition Action for the next step.



10. Rename it to **Check MDN Expected**. In 1st Choose a value box, select **MDNExpected** from **Decode AS2 Message** under **Dynamic Content**.



IsNrrEnabled
Gets or sets NRR Enabled.

MdnExpected
Gets or sets Acknowledgement expectation.

MdnStatusCode
Gets or sets Acknowledgement status code.

MdnStatusCode
Gets or sets Acknowledgement status code.

MdnType
Gets or sets the MDN type.

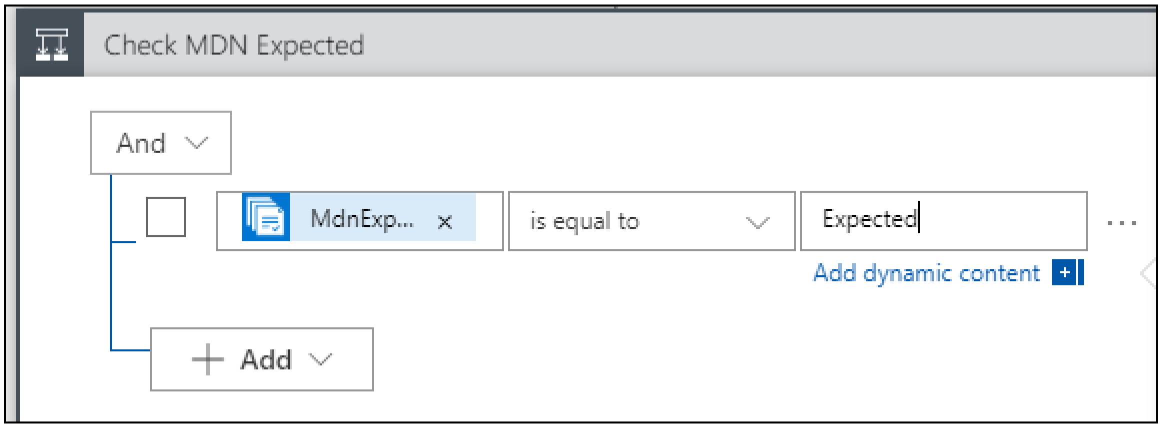
MicVerification
Gets or sets MIC Verification

MicVerification
Gets or sets MIC Verification

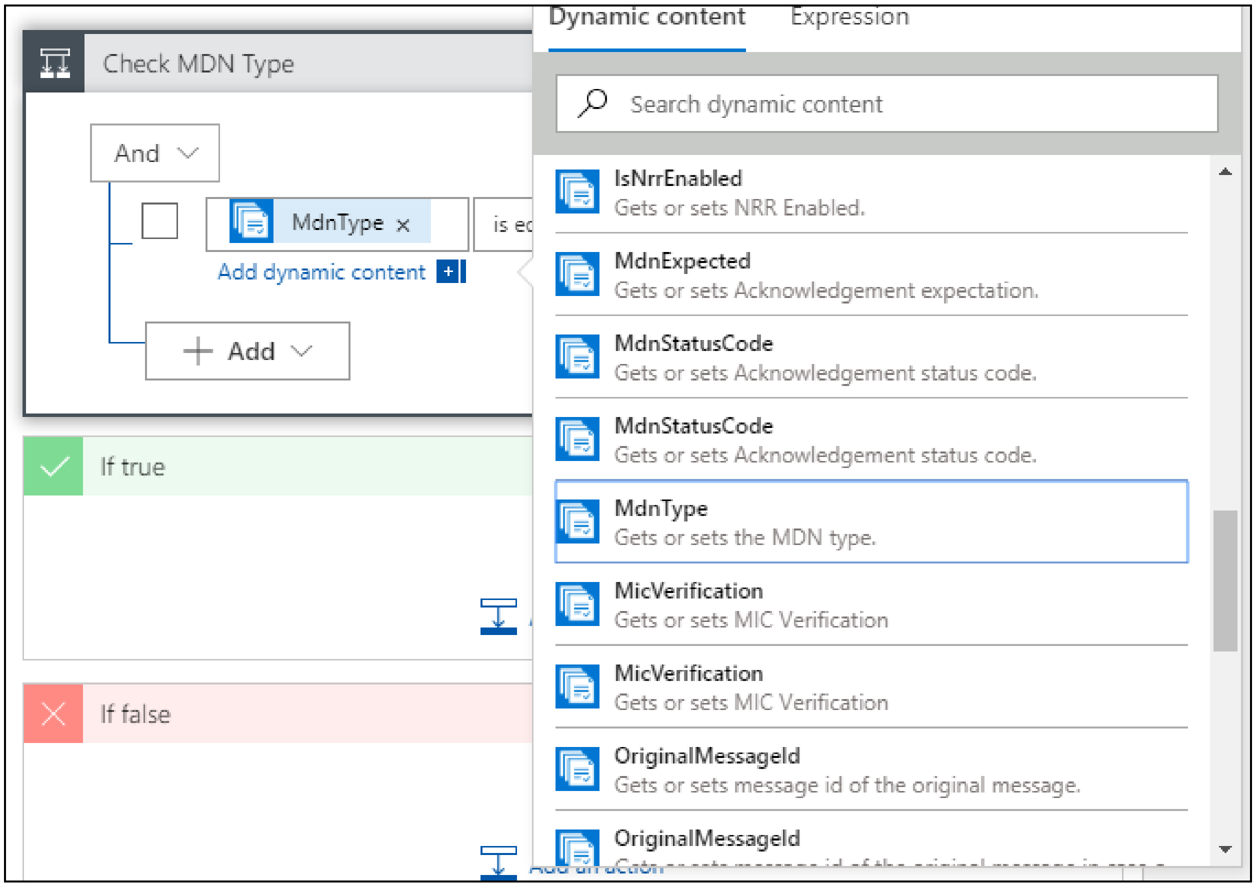
OriginalMessageId
Gets or sets message id of the original message.

OriginalMessageId

11. In the 2nd Choose a value box, type **Expected**. This is to check if the MDN is required by the Sender Partner.



12. In the True branch, add another Condition Action to check the MDN Type. Rename the Action to Check MDN Type. In 1st Choose a value box, select **MDNType** from **Decode AS2 Message** under **Dynamic Content**.



IsNrrEnabled
Gets or sets NRR Enabled.

MdnExpected
Gets or sets Acknowledgement expectation.

MdnStatusCode
Gets or sets Acknowledgement status code.

MdnStatusCode
Gets or sets Acknowledgement status code.

MdnType
Gets or sets the MDN type.

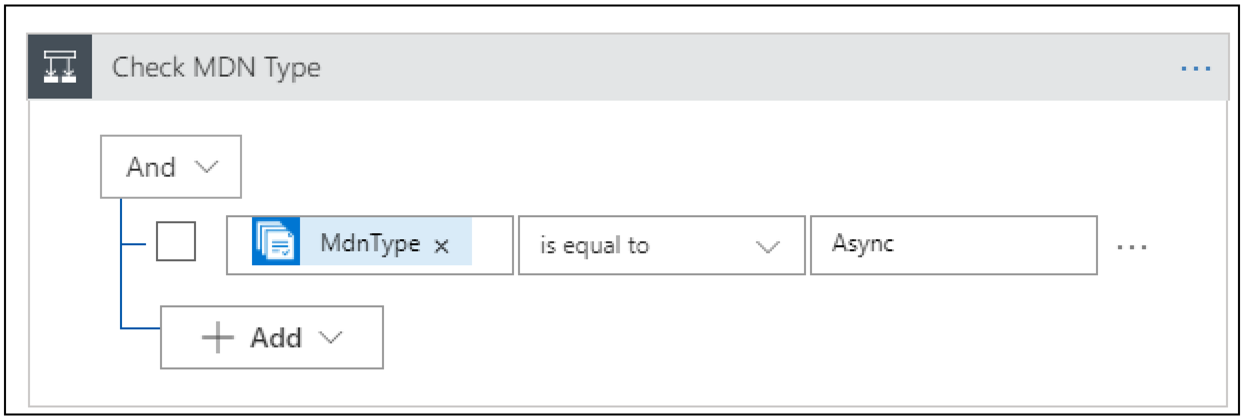
MicVerification
Gets or sets MIC Verification

MicVerification
Gets or sets MIC Verification

OriginalMessageId
Gets or sets message id of the original message.

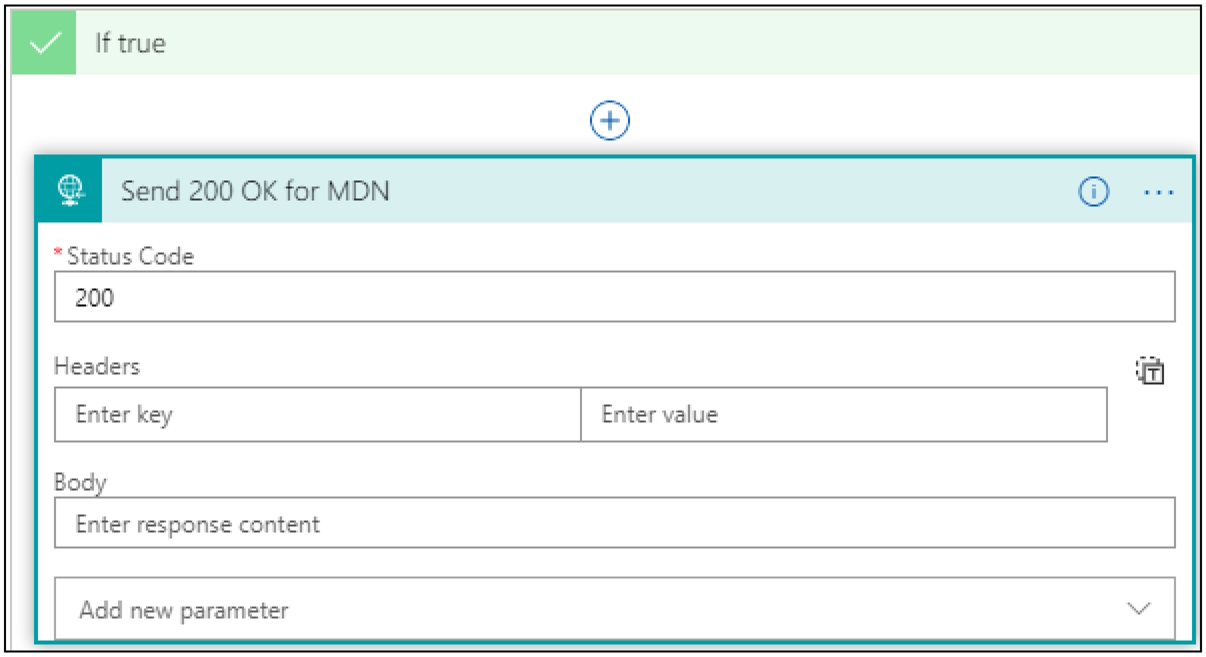
OriginalMessageId

13. In the 2nd Choose a value box, type **Async**. This is to check if the Asynchronous MDN has been requested



MdnType × | is equal to | Async

14. In the True branch, add Http Response Action. Rename it to **Send 200 OK for MDN**.


If true

Send 200 OK for MDN

* Status Code
200

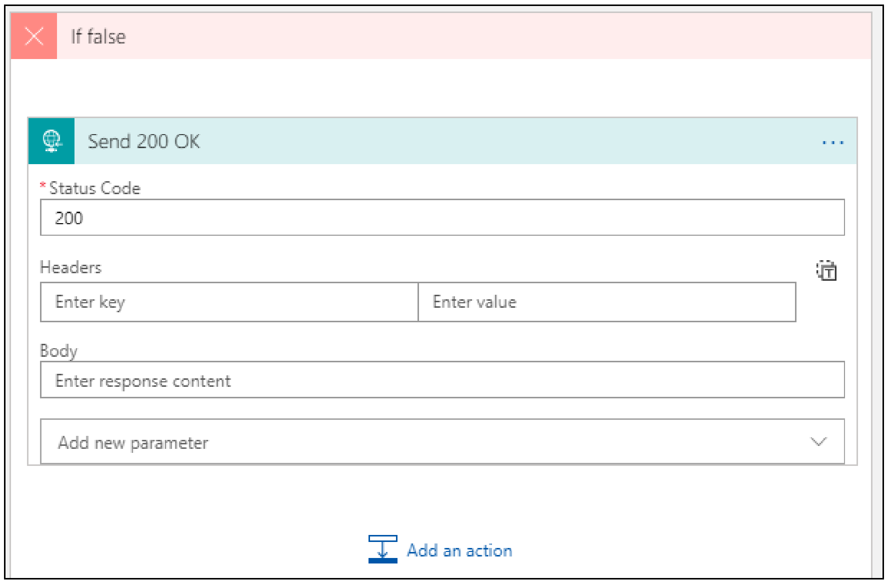
Headers
Enter key | Enter value

Body
Enter response content

Add new parameter

15. Leave the next False Branch blank this PoC scope includes handling of Asynchronous MDNs only. Synchronous MDN can be handled creating one more Logic App that is out of scope for this blog.

16. For the last False Branch (which is for Check MDN Expected condition), add Http Response Action. Rename it to **Send 200 OK**.


If false

Send 200 OK

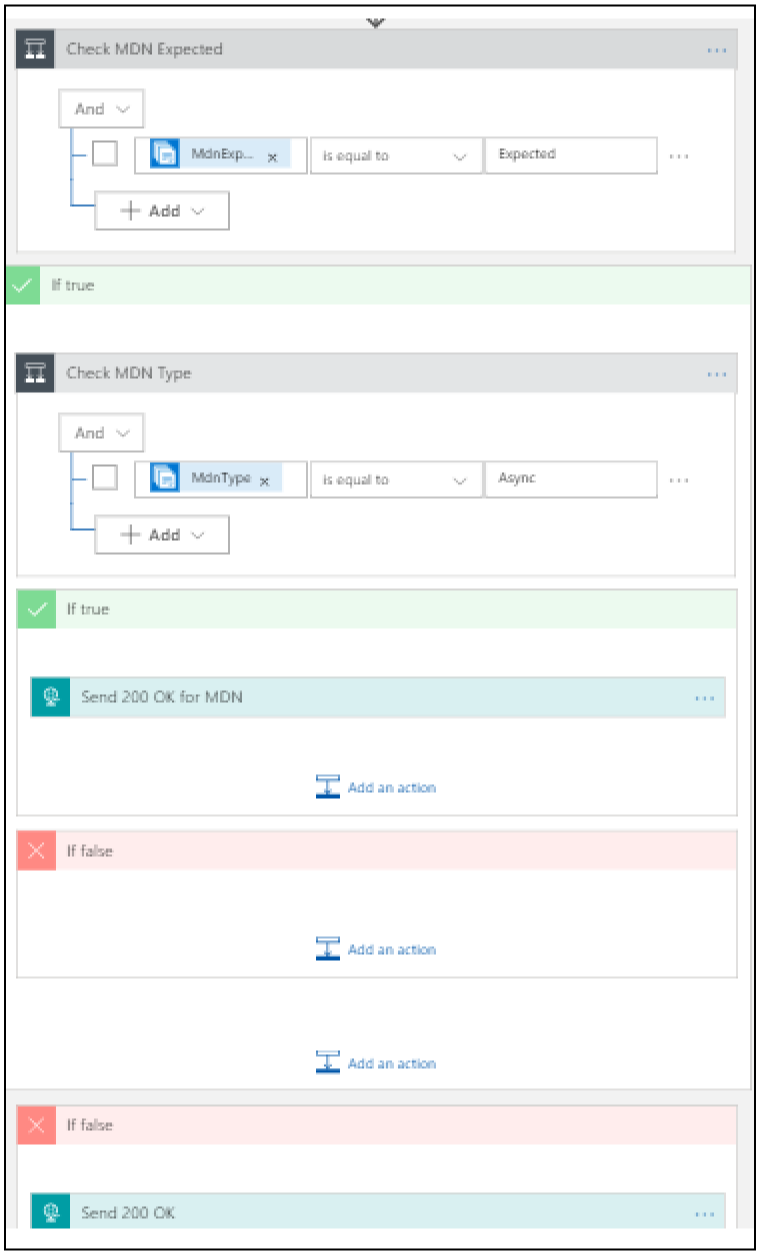
* Status Code
200

Headers
Enter key | Enter value

Body
Enter response content

Add new parameter

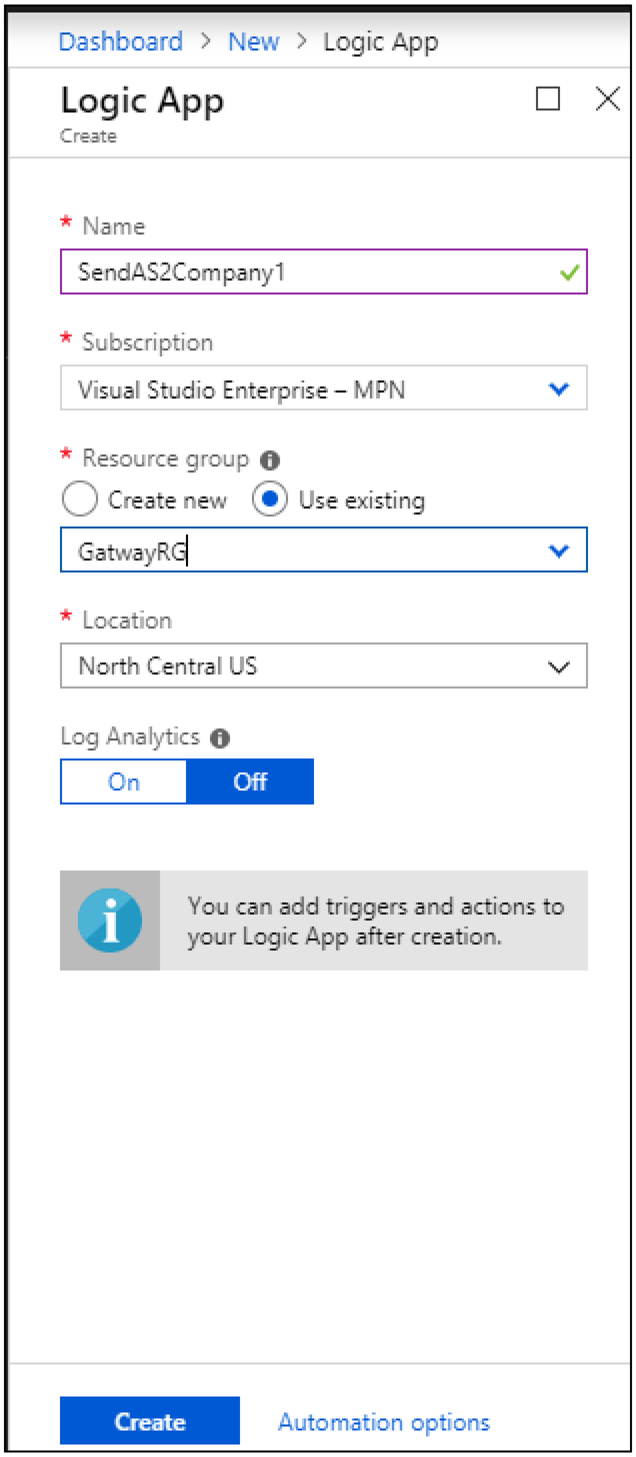
It looks like below.



17. Save the logic app.

Create Logic App to Send AS2 messages-

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



Dashboard > New > Logic App

Logic App
Create

* Name
SendAS2Company1

* Subscription
Visual Studio Enterprise – MPN

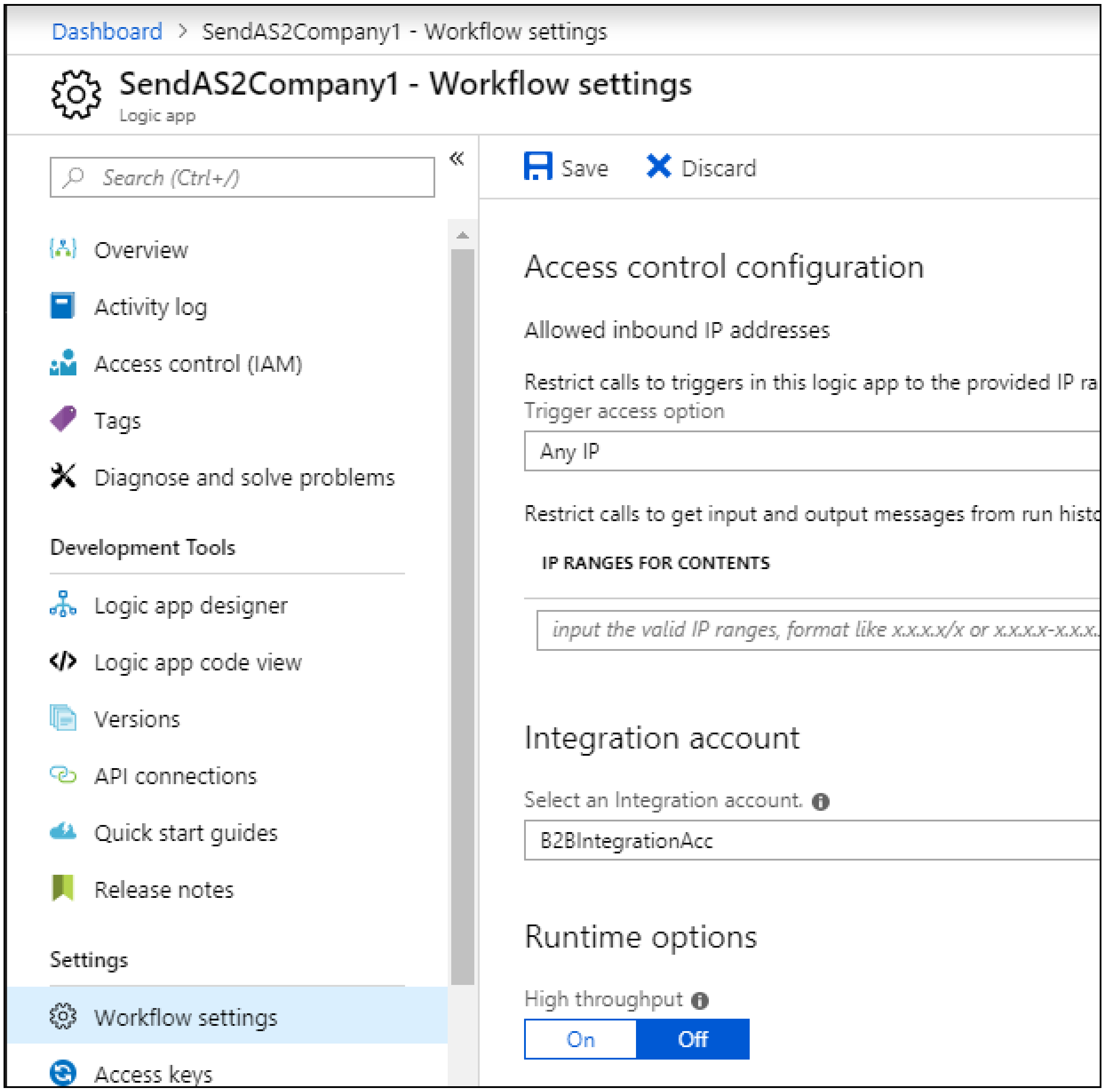
* Resource group
( ) Create new [✓ Use existing]
GatwayRG

* Location
North Central US

Log Analytics
On [✓ Off]

You can add triggers and actions to your Logic App after creation.

2. Link this logic app to your Integration Account.


Dashboard > SendAS2Company1 - Workflow settings

SendAS2Company1 - Workflow settings
Logic app

Access control configuration

Allowed inbound IP addresses

Restrict calls to triggers in this logic app to the provided IP ra
Trigger access option: Any IP

Restrict calls to get input and output messages from run histo

IP RANGES FOR CONTENTS

input the valid IP ranges, format like x.x.x.x/x or x.x.x.x-x.x.x.

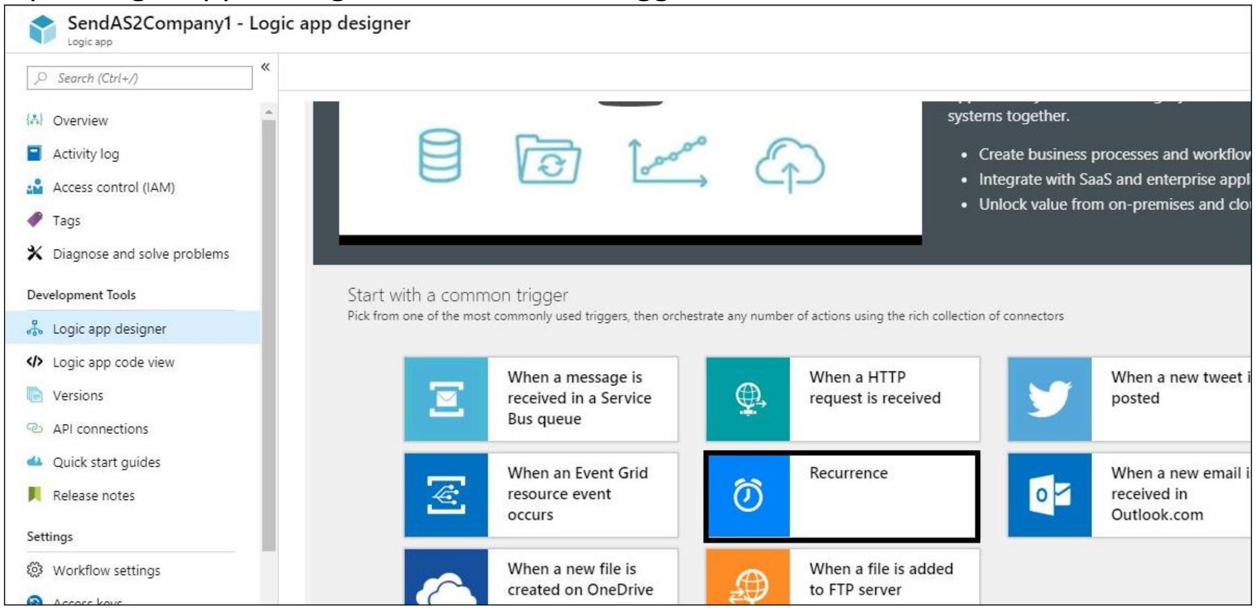
Integration account

Select an Integration account.: B2BIntegrationAcc

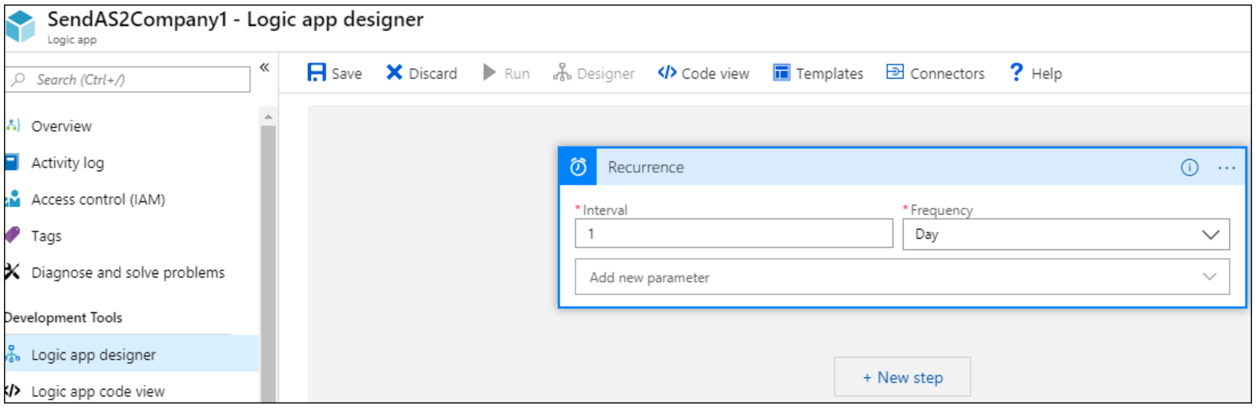
Runtime options

High throughput: On [✓ Off]

3. Open Logic Apps Designer. Choose the Trigger- **Recurrence**.

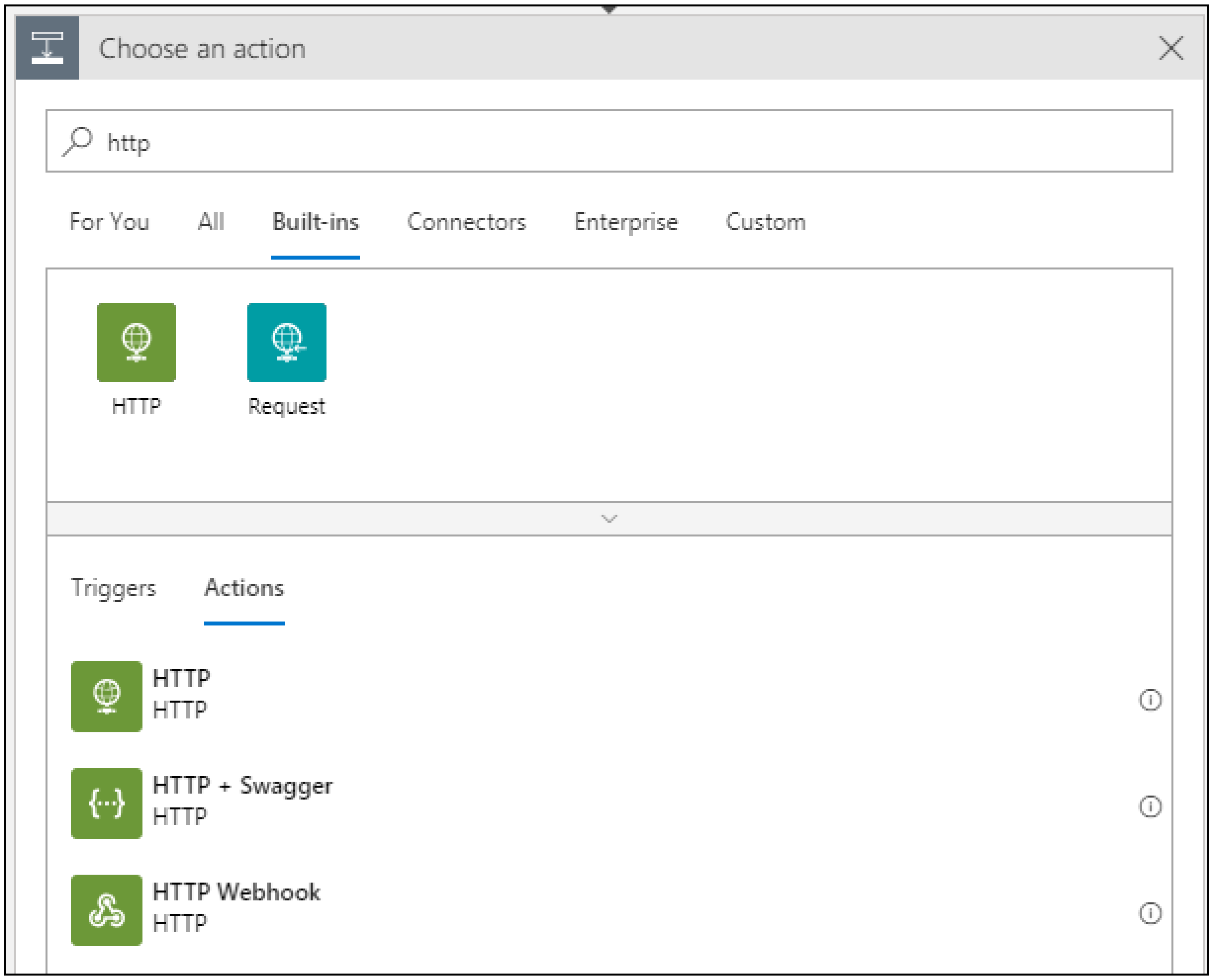

SendAS2Company1 - Logic app designer
Logic app
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Development Tools
Logic app designer
Logic app code view
Versions
API connections
Quick start guides
Release notes
Settings
Workflow settings
systems together.
Create business processes and workflow
Integrate with SaaS and enterprise appl
Unlock value from on-premises and clo
Start with a common trigger
Pick from one of the most commonly used triggers, then orchestrate any number of actions using the rich collection of connectors
When a message is received in a Service Bus queue
When a HTTP request is received
When a new tweet i posted
When an Event Grid resource event occurs
Recurrence
When a new email i received in Outlook.com
When a new file is created on OneDrive
When a file is added to FTP server

4. Define the Interval and the frequency.



*Interval: 1
*Frequency: Day

5. Add new step and search with **http**. Under the **Built-ins**, select **HTTP**.


Choose an action

http

For You · All · Built-ins · Connectors · Enterprise · Custom

HTTP · Request

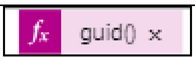
Triggers · Actions

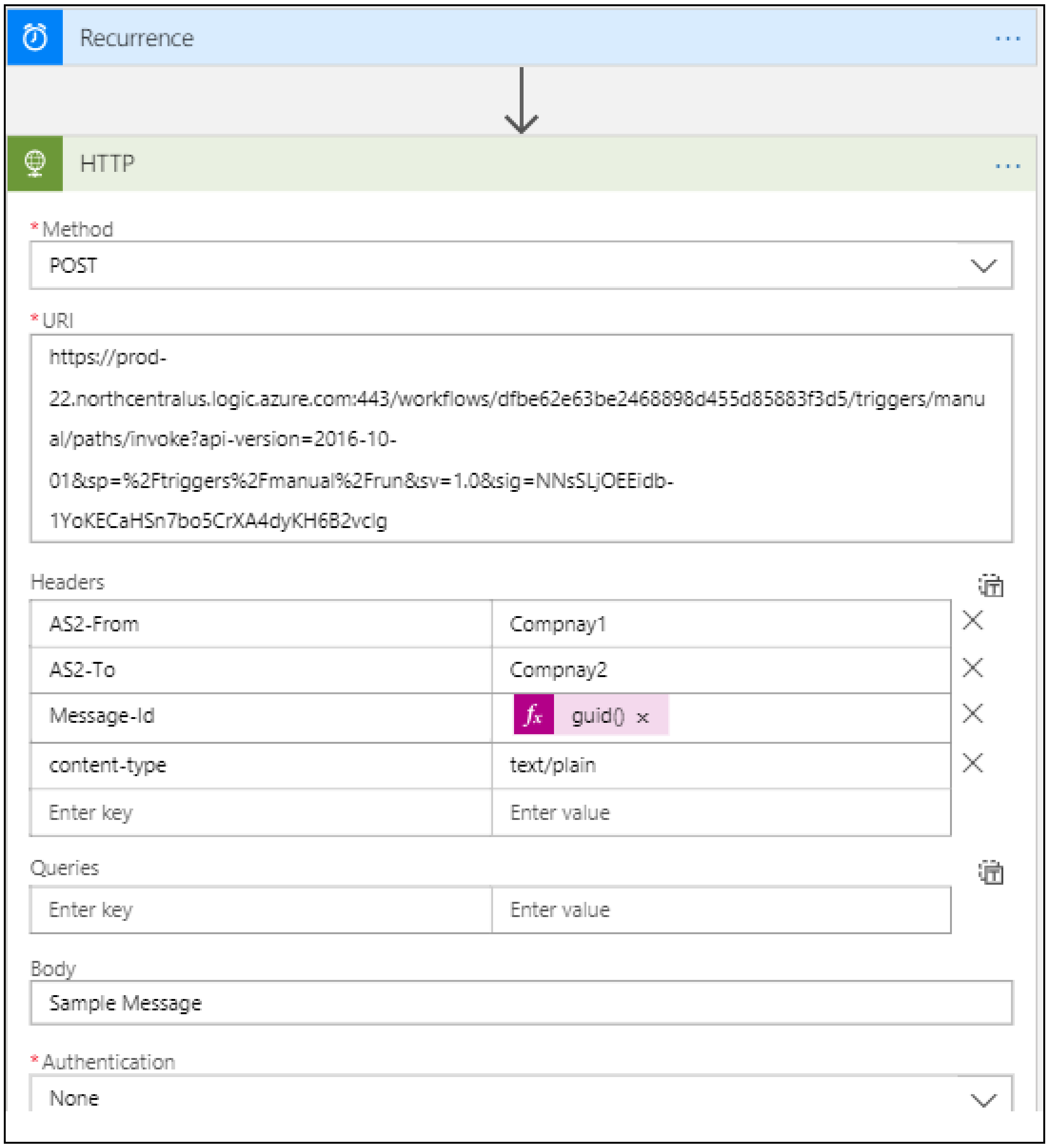
HTTP
HTTP

HTTP + Swagger
HTTP

HTTP Webhook
HTTP

6. In the HTTP Action, configure as below:

Method	Post
URL	The Trigger URL of Receive Logic App which you made a note of in step 5 of creating Receive Logic App.
Headers-	
AS2-From	Company1
AS2-To	Company2
Message-Id	 use guid() function to auto generate GUID.
content-type	text/plain
Body-	Sample Message
Authentication	None



* Method
POST

* URI
https://prod-22.northcentralus.logic.azure.com:443/workflows/dfbe62e63be2468898d455d85883f3d5/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=NNsSLjOEEidb-1YoKECaHSn7bo5CrXA4dyKH6B2vclg

Headers

AS2-From	Compnay1
AS2-To	Compnay2
Message-Id	guid()
content-type	text/plain
Enter key	Enter value

Queries

Enter key	Enter value

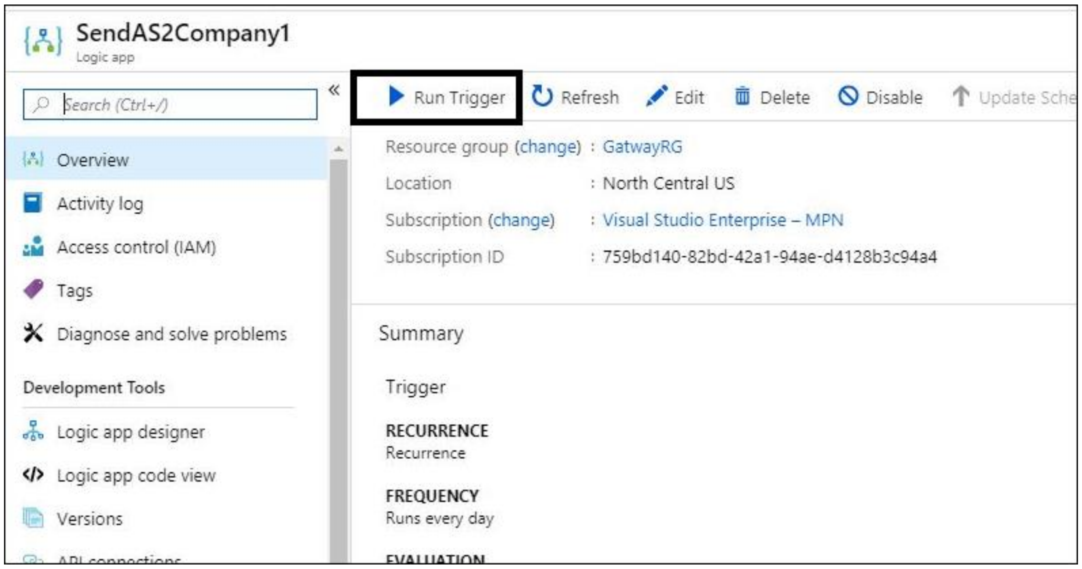
Body
Sample Message

* Authentication
None

7. Save the Logic App.

Test the Solution-

1. Open the **SendAS2Company1** Logic App blade and click on Run Trigger to run it.


SendAS2Company1
Logic app

Resource group (change) : GatwayRG
Location : North Central US
Subscription (change) : Visual Studio Enterprise – MPN
Subscription ID : 759bd140-82bd-42a1-94ae-d4128b3c94a4

Summary

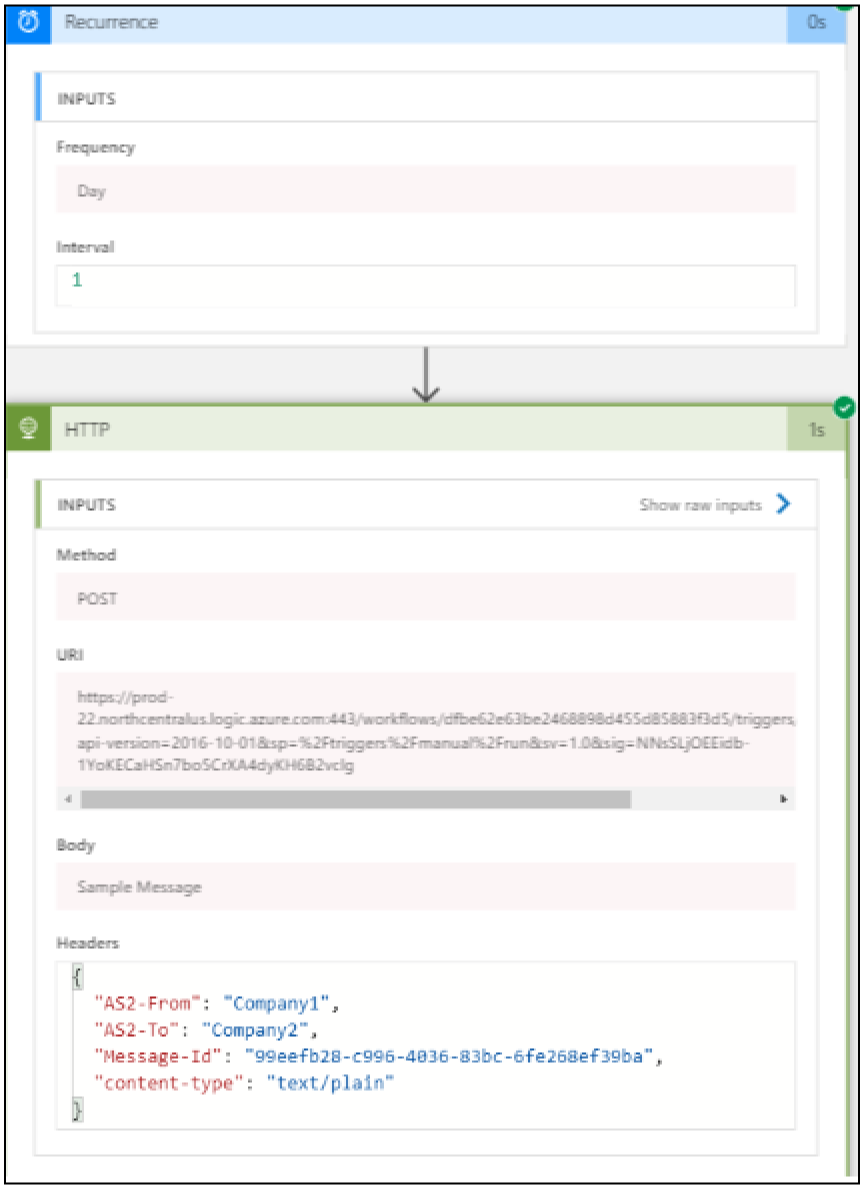
Trigger

RECURRENCE
Recurrence

FREQUENCY
Runs every day

EVALUATION

2. Check the run history and input/output for each action for these logic apps.
   SendAS2Company1 Logic app-


Recurrence 0s

INPUTS

Frequency

Day

Interval

1

HTTP 1s

INPUTS

Method

POST

URI

```
https://prod-
22.northcentralus.logic.azure.com:443/workflows/dfbe62e63be2468898d455d85883f3d5/triggers,
api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=NNsSLjOEEidb-
1YoKECaHSn7boSCrXA4dyKH6B2vclg
```

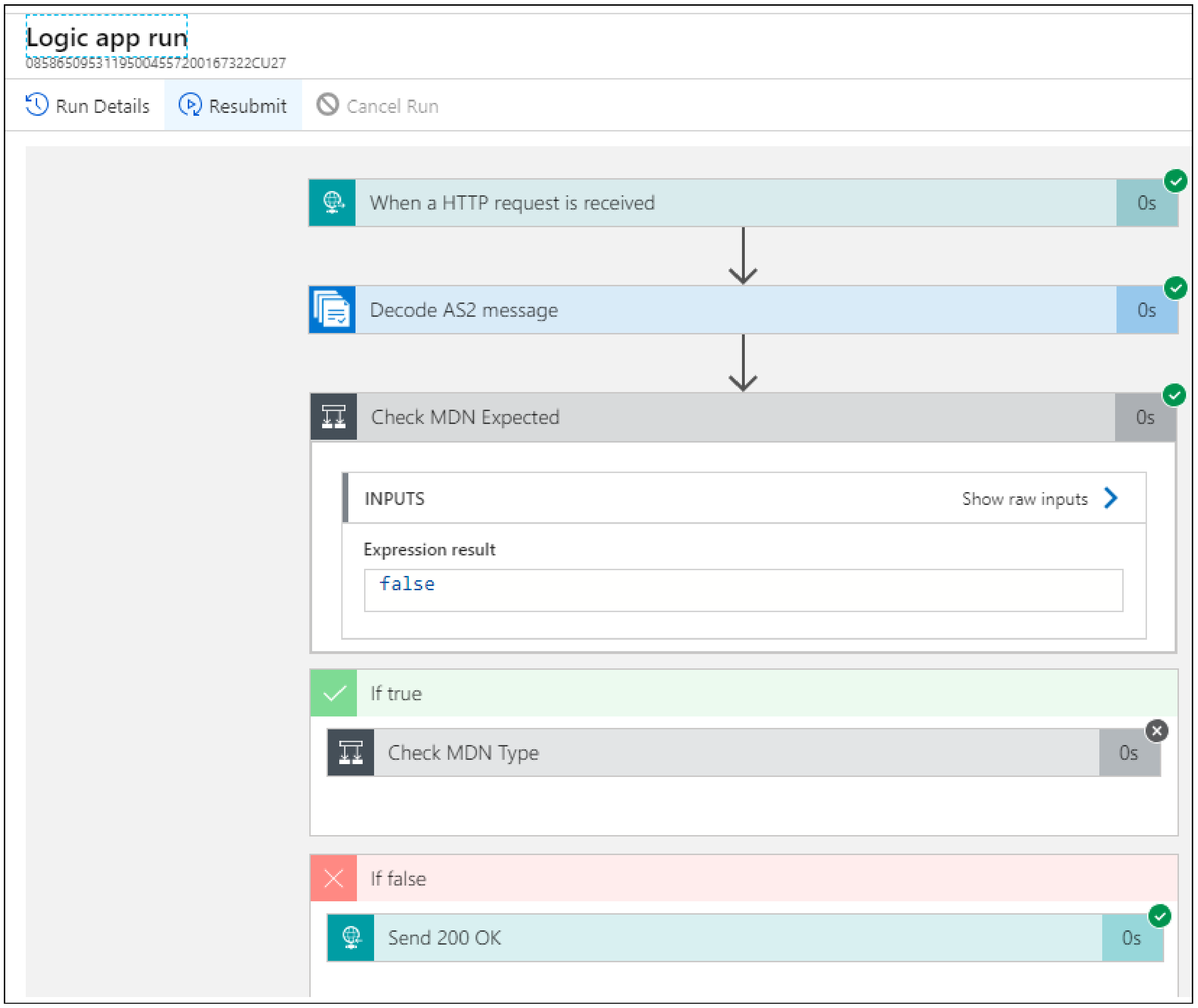
Body

Sample Message

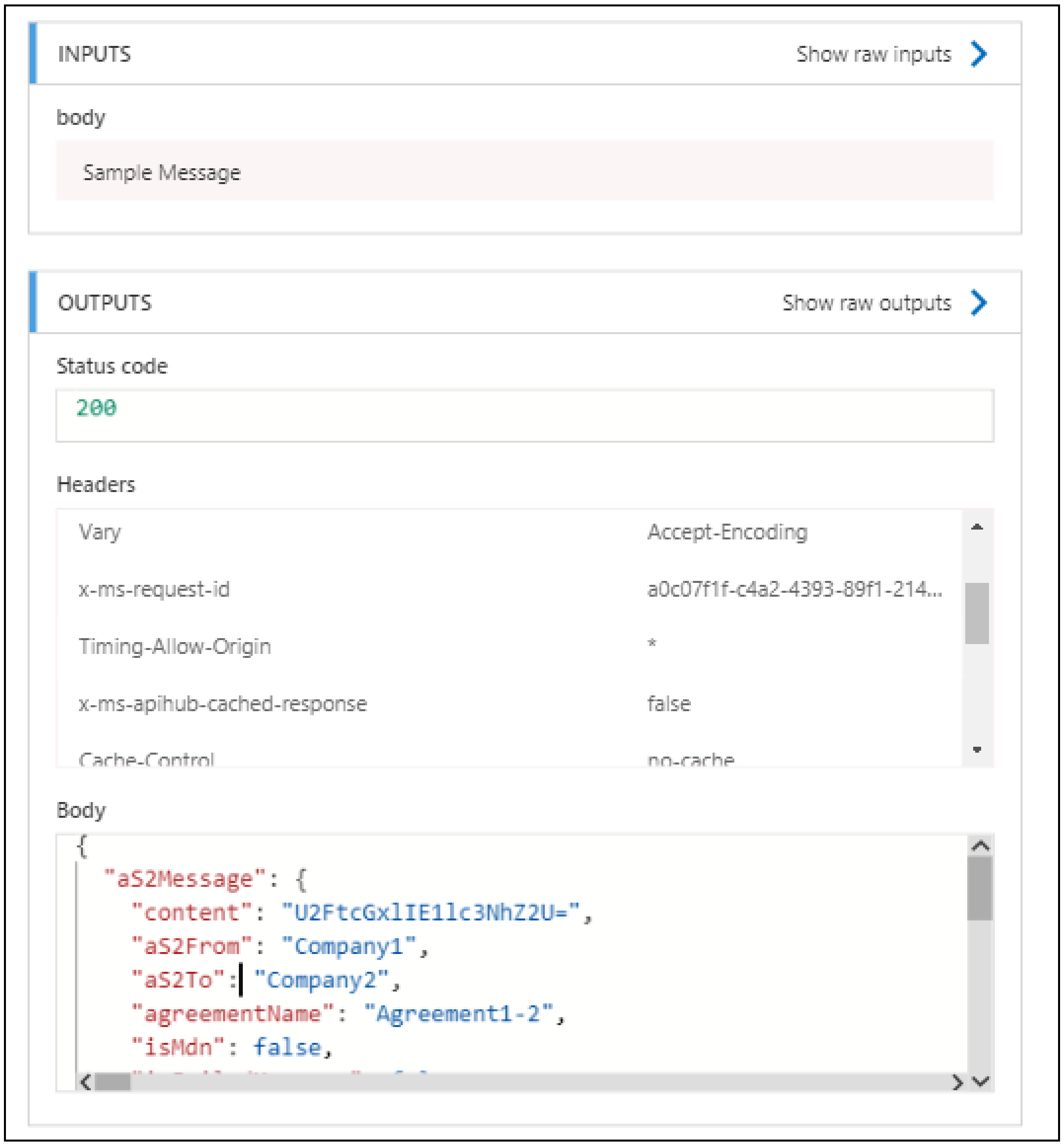
Headers

```
{
  "AS2-From": "Company1",
  "AS2-To": "Company2",
  "Message-Id": "99eefb28-c996-4036-83bc-6fe268ef39ba",
  "content-type": "text/plain"
}
```

Receive AS2Company2 Logic App (It will receive the AS2 message sent by **SendAS2Company1** Logic App)-



Decode AS2 Message Action-


INPUTS Show raw inputs

body

Sample Message

OUTPUTS Show raw outputs

Status code

200

Headers

Vary	Accept-Encoding
x-ms-request-id	a0c07f1f-c4a2-4393-89f1-214...
Timing-Allow-Origin	*
x-ms-apihub-cached-response	false
Cache-Control	no-cache

Body

```
{
  "aS2Message": {
    "content": "U2FtcGxlIE1lc3NhZ2U=",
    "aS2From": "Company1",
    "aS2To": "Company2",
    "agreementName": "Agreement1-2",
    "isMdn": false,
```

You will see that it has resolved the agreement we had created to receive the message and all the relevant information is received in message body.

AS2 decoder details-

The Decode AS2 connector performs these tasks:

- Processes AS2/HTTP headers
- Verifies the signature (if configured)
- Decrypts the messages (if configured)
- Decompresses the message (if configured)

- Check and disallow message ID duplicates (if configured)
- Reconciles a received MDN with the original outbound message
- Updates and correlates records in the non-repudiation database
- Writes records for AS2 status reporting
- The output payload contents are base64 encoded
- Determines whether an MDN is required, and whether the MDN should be synchronous or asynchronous based on configuration in AS2 agreement
- Generates a synchronous or asynchronous MDN (based on agreement configurations)
- Sets the correlation tokens and properties on the MDN