Bachelor's thesis

Information and Communications Technology

2022

Juho Piispanen

Microsoft Azure as an Integration Platform

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Bachelor’s Thesis | Abstract

Turku University of Applied Sciences

Information and Communications Technology

2022 | 28 pages

Juho Piispanen

Microsoft Azure as an Integration Platform

Application integration is an essential part of getting the most from software in use. Most of the time organizations cannot rely on only one application to do everything, so connecting two or more applications is necessary. Organizations are moving services increasingly from on-premises datacenters into the public cloud and are using cloud-based integration platforms rather than traditional integration technologies such as BizTalk Servers. These integration Platform as a Service (iPaaS) solutions such as Microsoft Azure Integration Services are a set of cloud services for mission critical enterprise integration.

Objective for this thesis is to take a closer look at Microsoft Azure Integration Services, learn the advantages of these modern iPaaS services and how to use them in practice.

The thesis summarizes the core concepts of Azure Integration Services (API Management, Service Bus, Logic Apps ja Event Grid) and using some of them in practice. At the end of the thesis, Azure Logic Apps integration is built, and the steps and tools are presented.

Keywords:

Microsoft Azure, Logic Apps, API Management, Service Bus, Event Grid, Integration Platform as a Service

Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2022 | 28 sivua

Juho Piispanen

Microsoft Azure integraatioalustana

Järjestelmäintegraatiot ovat olennainen osa organisaation toimintaa, jotta ohjelmistoista saataisiin mahdollisimman paljon irti. Suurin osa organisaatioista käyttää useampaa eri ohjelmistoa, joten ohjelmistojen yhdistäminen on tarpeellista. Yritykset ovat vaihtamassa integraatiopalvelujaan paikallisista datakeskuksista yhä enemmän ja enemmän julkiseen pilveen perinteisten integraatioteknologioiden, kuten BizTalk palvelimien sijaan. Nämä integration Platform as a Service (iPaaS) ratkaisut, kuten Microsoft Azuren integraatiopalvelut, ovat kokoelma pilvipalveluja tehtäväkriittisille yritysintegraatioille.

Tämän opinnäytetyön tavoitteena oli perehtyä Microsoft Azuren integraatiopalveluihin, oppia integration Platform as a Service teknologioiden hyödyt sekä niiden hyödyntäminen käytännössä.

Opinnäytetyössä tehtiin yhteenveto keskeisistä Microsoft Azuren integraatiopalveluista (API Management, Service Bus, Logic Apps ja Event Grid) käytännön esimerkein. Opinnäytetyön lopussa rakennetaan Azure Logic Apps integraatio, jonka keskeiset vaiheet ja käytetyt työkalut esitellään.

Asiasanat:

Microsoft Azure, Logic Apps, API Management, Service Bus, Event Grid, Integration Platform as a Service

Content

Pictures

List of abbreviations

API	Application Programming Interface
JSON	JavaScript Object Notation
API	Application Programming Interface
iPaaS	Integration Platform as a Service
CLI	Command Line Interface
SDK	Software Development Kit
HTTP	Hypertext Transfer Protocol

1 Introduction

Microsoft Azure is a mainly pay-as-you-go set of cloud services for modern businesses. Azure gives the freedom to manage, build and deploy applications with various tools and frameworks and user pays for only what they use. [2] Azure provides for example compute, networking, storage and integration services. [3]

The public cloud has many benefits compared to on-premises datacenters. Cloud computing saves companies money since companies don't have to buy hardware or software. Cloud computing also adds scalability. It is easy and fast to get more for example computing power and storage from cloud providers when needed without needing to set anything up. Cloud providers also offer a wide variety of security to help protect the companies' data. [14]

Businesses are increasingly starting to migrate from traditional on-premises data centers into the public cloud. This change includes integration technologies. Companies change traditional integration services to iPaaS solutions such as Microsoft Azure Integration Services. [1]

There are different providers for iPaaS solutions for example Microsoft, IBM, MuleSoft and Boomi. They all provide a different environment and tools for creating integration services, but they all have the same basic idea. This thesis will concentrate to Microsoft's technologies. [15]

There are four components of Azure Integration Services [1]:

- API Management
- Logic Apps
- Service Bus
- Event Grid

The thesis is divided into four different chapters. Chapter 1 provides an introduction to Microsoft Azure and Azure Integration Services.

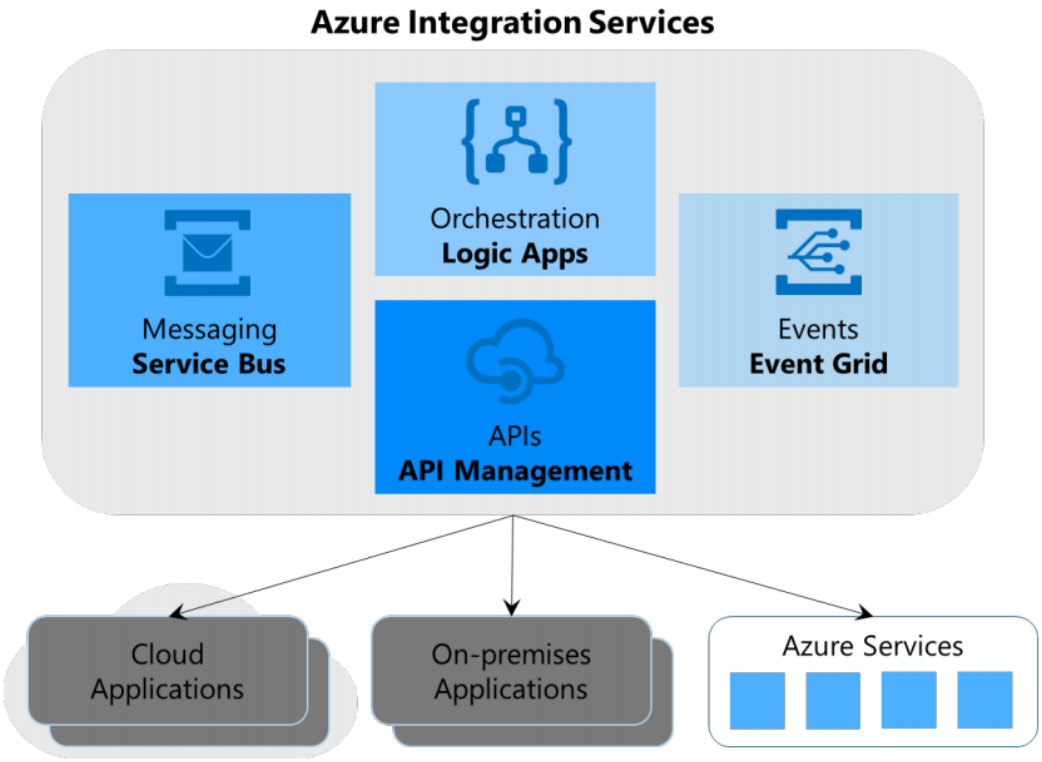
Chapter 2 will dive a little deeper to Azure Integration Services explaining each services functionality, how to use them and why it is a clever idea to combine them with each other.

Chapter 3 will display the different components and tools used as a part of the example implementation done in the last chapter. The tools and components will be described in detail.

In chapter 4, a Logic App will be created using the tools described in detail in chapter 3. Chapter 4 will cover the development process of Logic Apps and show what to do with them.

2 Azure Integration Services Components

Azure Integration Services has four main components which each have a different purpose. The full potential of the components becomes clear when they are all used together. An example scenario: API management takes in request, Service Bus takes in the message and sends it to Event Grid, Event Grid creates an event which triggers a Logic App that executes a business process. [1]



Picture 1. Azure Integration Services components. [16]

2.1 API Management

Azure API Management is a hybrid, multi-cloud management platform for APIs. The components are Azure-hosted and fully managed by default. Azure API Management is made up of three main components [4]:

- API Gateway
- Management Plane
- Developer Portal

The API Gateway takes in all requests from client applications and forwards them to backend services. The gateway enables consistent configuration of security, routing, throttling, caching and observability. [4]

Management Plane allows API providers to interact with the service and provides full access to the API Management service capabilities. Management Plane is interacted through either Azure portal, Azure PowerShell, Azure CLI, Visual Studio Code extension or client SDKs by customers. [4]

Developer Portal is an open-source automatically generated and fully customizable website with the documentation of the APIs. API documentation can be read, APIs can be called via interactive console and API keys are managed via the Developer Portal. [4]

2.2 Service Bus

Azure Service Bus is an enterprise message broker with queues and publish-subscribe topics that is fully managed. Service Bus has many advantages for example Load-balancing, safe routing and transferring data and control and coordinating high-degree reliability transactional work. Service Bus integration to other Azure services such as Event Grid, Logic Apps and Azure Functions is seamless. Basic concepts of Service Bus are [5]:

- **Queues:** Messages are sent and received in queues, ordered and timestamped on arrival.
- **Topics:** Topics can be used to send and receive messages. Topics are useful in subscribe/publish scenarios.
- **Namespaces:** Namespaces often serve as application containers for all messaging components.

Service Bus comes with advanced features for complex messaging problems for example [5]:

- **Auto-forwarding:** Enables to chain a queue to another queue.
- **Scheduled delivery:** Schedule messages to become available for processing at a certain time.
- **Auto-delete on idle:** Delete queue after a specified interval.
- **Security:** Supports standard Advanced Message Queuing Protocol (AMQP) 1.0 and HTTP/Rest Protocols.

2.3 Logic Apps

Azure Logic Apps is a low-code/no-code solution for creating and running automated workflows to integrate apps, data, services and systems. Logic Apps make it simple for example to connect legacy systems to modern systems across cloud, on premises and hybrid environments. Basic Logic App concepts are [6]:

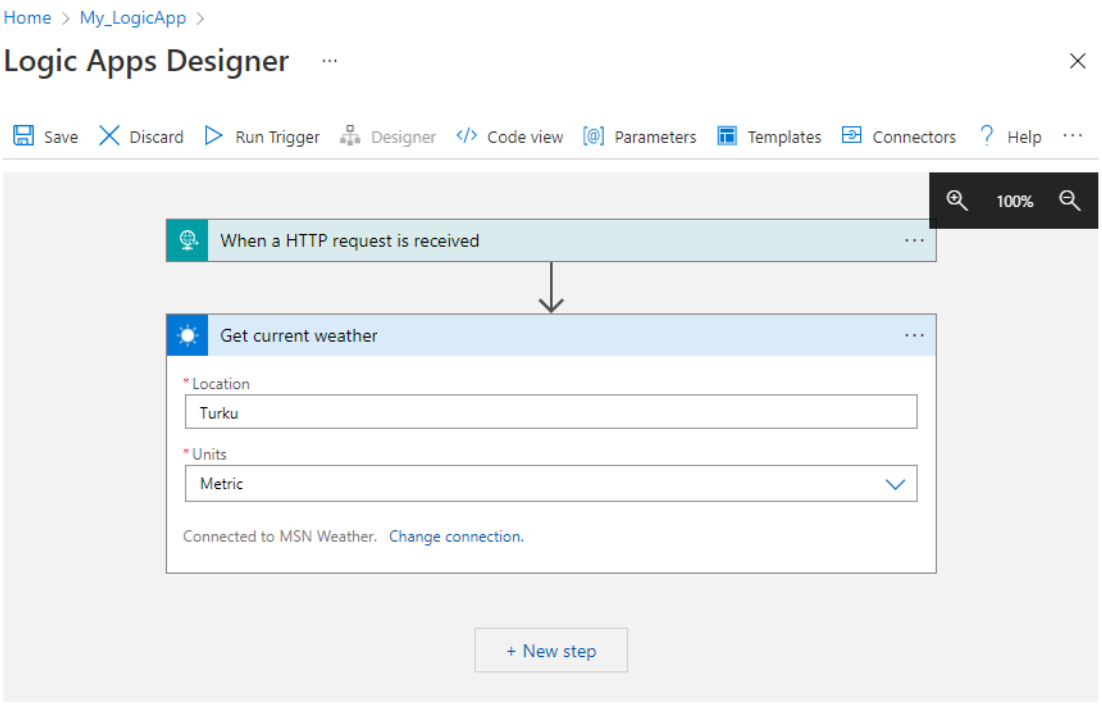
- **Workflow:** The body of the Logic App on top of which the app is created. Workflow starts with a trigger followed by one or more actions.
- **Trigger:** Is the first step of a workflow which specifies the condition for running actions in the workflow.
- **Action:** Each step after trigger is an action in the workflow.

Logic Apps fall into two categories: Consumption and Standard. Both types have their advantages and disadvantages. Consumption model Logic Apps are fully managed and easier to get started with, but a single Logic App can have only one workflow. Standard Logic Apps are a little more complex and can have multiple stateful and stateless workflows in a single Logic App. [6]

Logic Apps have built-in Microsoft-managed API connectors for integrating apps, data, services and systems easily and fast. Logic Apps are a low-

code/no-code solution which means that it is easy to get started without much coding experience. Sometimes code needs to be written for custom actions and that can be done with Azure Functions or adding an Inline Code action. [6]

Logic Apps can be created via Azure Logic Apps designer in Azure Portal, Visual Studio or Visual Studio Code with ready templates to take advantage of. [6]



Picture 2. Logic Apps Designer with a trigger and an action.

2.4 Event Grid

Azure Event Grid allows easy building of applications with event-based architectures. Event Grid works by selecting an Azure resource to subscribe to and giving event handler or Webhook endpoint to send it to. Event Grid can be used with built-in Azure services or custom topics. [7]

There are many Azure services that support sending events to Event Grid for example [7]:

- Azure API Management
- Azure IoT Hub
- Azure Key Vault

There are also many Azure Services that support handling events for example [7]:

- Azure Functions
- Logic Apps
- Service Bus

Event Grid can be used to connect apps with other services. Event Grid works with Logic Apps flawlessly. Set Event Grid as trigger, subscribe to a resource event and event handler triggers the Logic App. [7]

3 Tools and Technologies

Azure resources must be located in resource groups. Resource group is a container in Azure containing related resources for an Azure solution. User can choose whether to have all resources of a solution in the same resource group or rather only the resources that are wanted to be managed together. [13]

3.1 Azure Portal

In chapter four, a Logic App (Consumption) will be defined, implemented and deployed. Logic Apps can be made in either Azure Portal, Microsoft Visual Studio or Microsoft Visual Studio Code. Azure Portal can be accessed via a web browser and has an easy-to-use graphical interface, so Azure Portal will be used to create the Logic App. Azure Portal is a hub which unifies applications including web apps, databases, virtual machines, virtual networks, storage and Visual Studio projects. It can also be used as a command-line interface provided by Cloud Shell. A registered Azure account is needed and an Azure subscription. [8]

3.2 Trigger

A HTTP trigger will be created, and it will be invoked with a payload containing cities from Finland in a JSON format. There are many tools for invoking a HTTP trigger and Postman will be used. Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration for creating better APIs. [9]

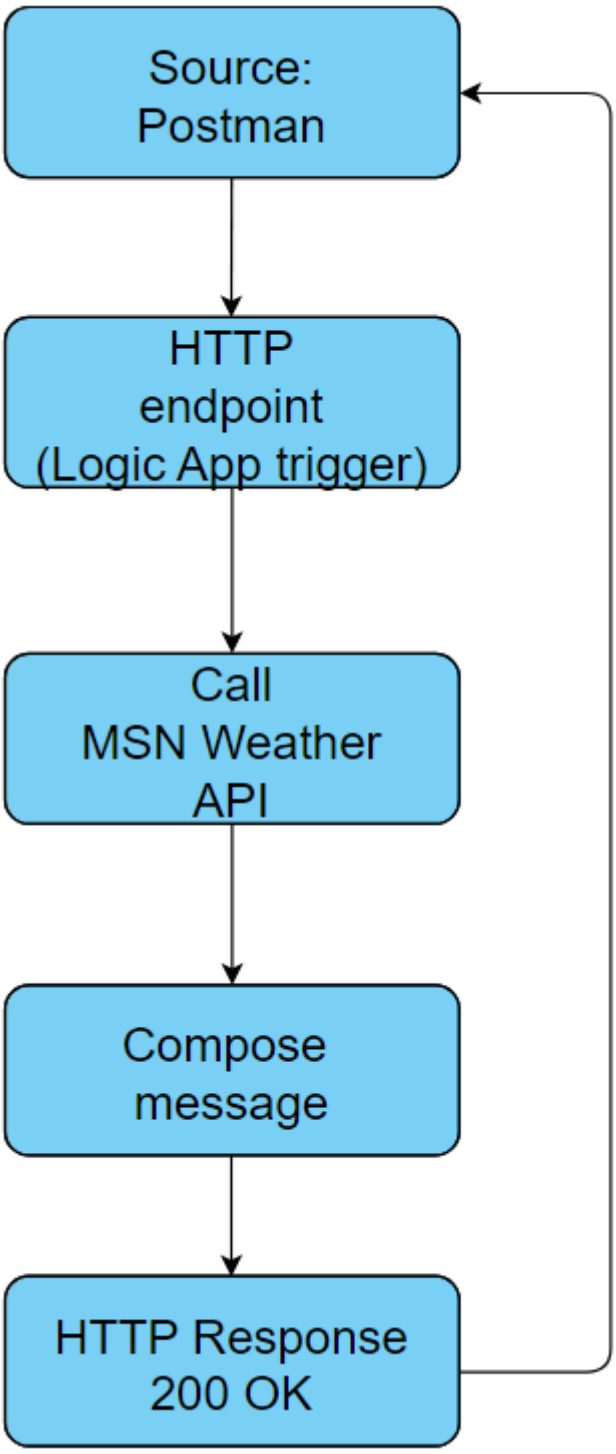
3.3 Actions

Actions need to be added to the Logic App for it to work. The idea of the Logic App is to look for the current weather data of each city in the request payload

and get the weather data as response. Built-in MSN Weather API will be used to get the weather data. [6]

To get the data for each city in the request payload, a loop needs to be made. For this purpose, a “Foreach” loop will work fine. A “Foreach” loop repeats one or more actions on each array item. It works only on arrays, so the payload must be in an array. Fortunately, JSON and arrays work together fine. [11]

Only specific data is wanted from the MSN Weather API so some data operations have to be made. Compose action can create a message or string from multiple inputs which can have various data types. [12]

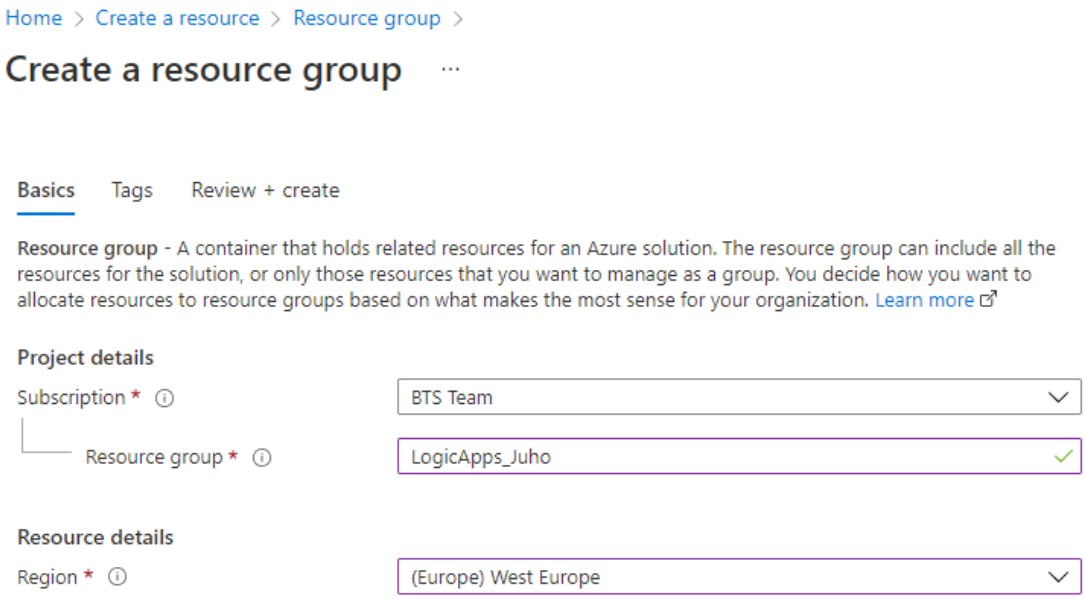


Picture 3. Sequence diagram for the Logic App.

4 Defining and Deploying a Logic App

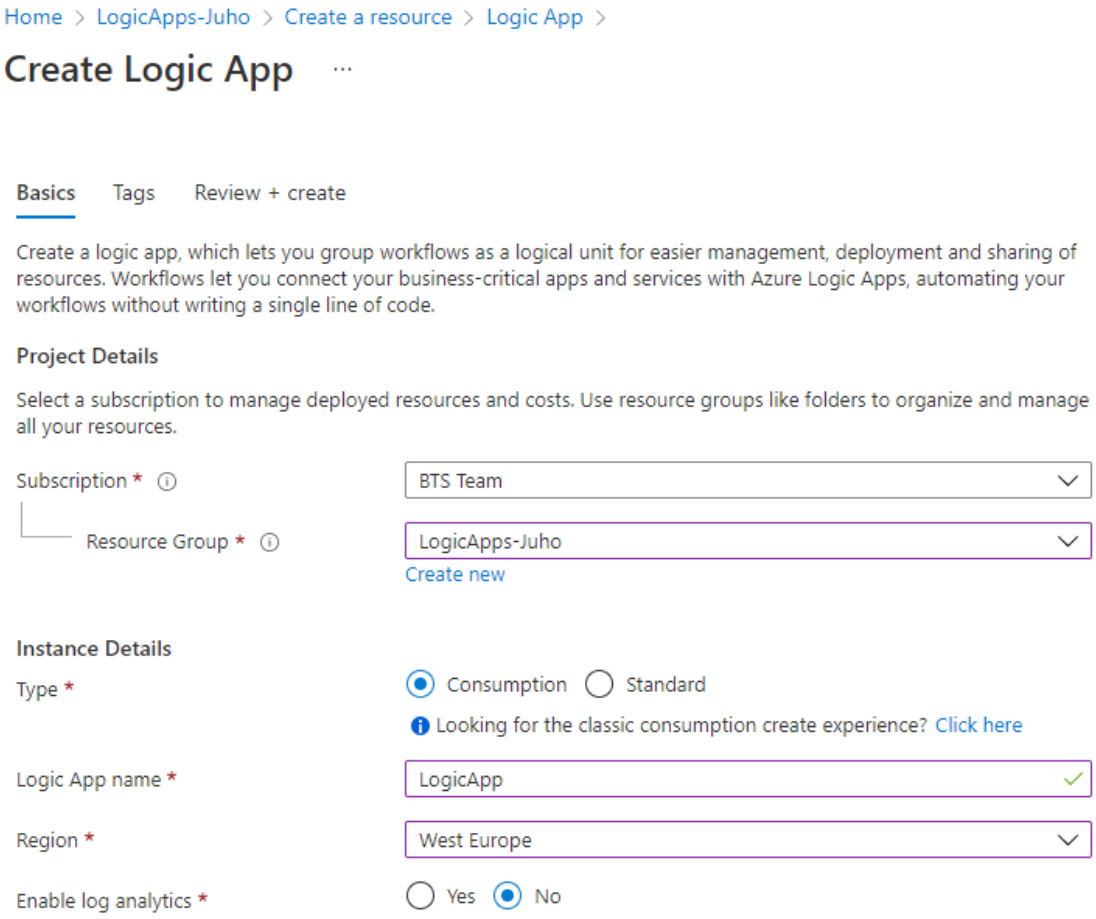
4.1 Setting up the environment

As stated before, a resource group must be made for all the resources of the solution. This happens in Azure Portal by clicking “Create a resource” and selecting “Resource group”. The resource group is given a name and a region.



Picture 4. Creating a resource group.

The Logic App is created inside the resource group by clicking “Create” and selecting “Logic App”. Type is set as “Consumption” and the Logic App is given a name and a region. It is a good practice to set the same region for each resource related to each other. West Europe in selected in this case.

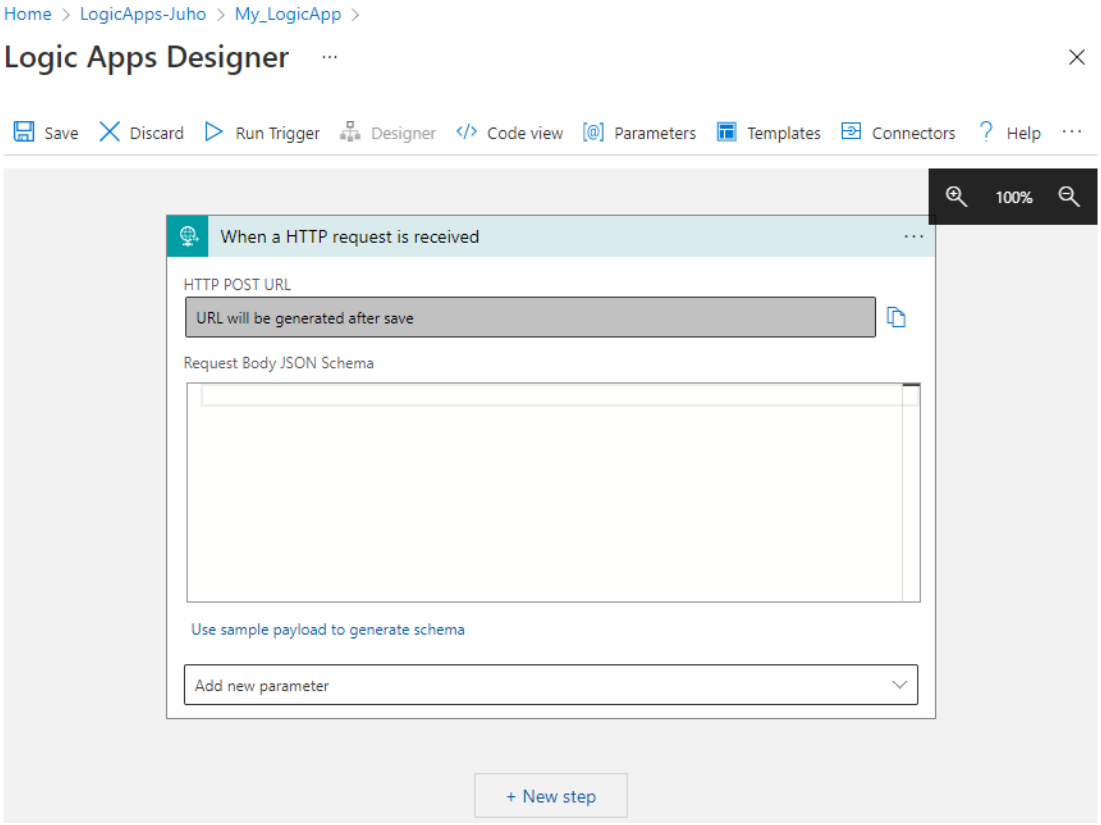


Picture 5: Creating a Logic App.

A resource group is now made for all resources and a Logic App is created.

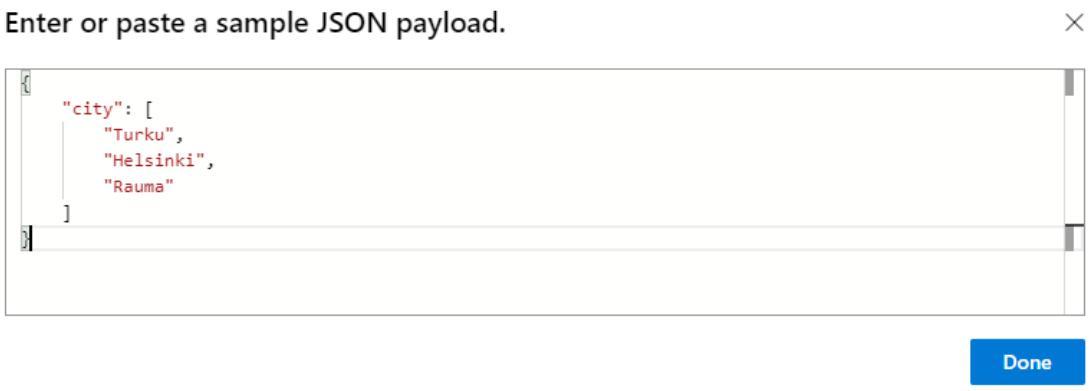
4.2 Getting started with the Logic App

When the new Logic App is opened, Azure gives ready templates for the Logic App. This time the ready templates are skipped, and “Blank Logic App” is selected. The trigger needs to be set first. “When a HTTP request is received” is selected.



Picture 6. Adding a trigger to the Logic App.

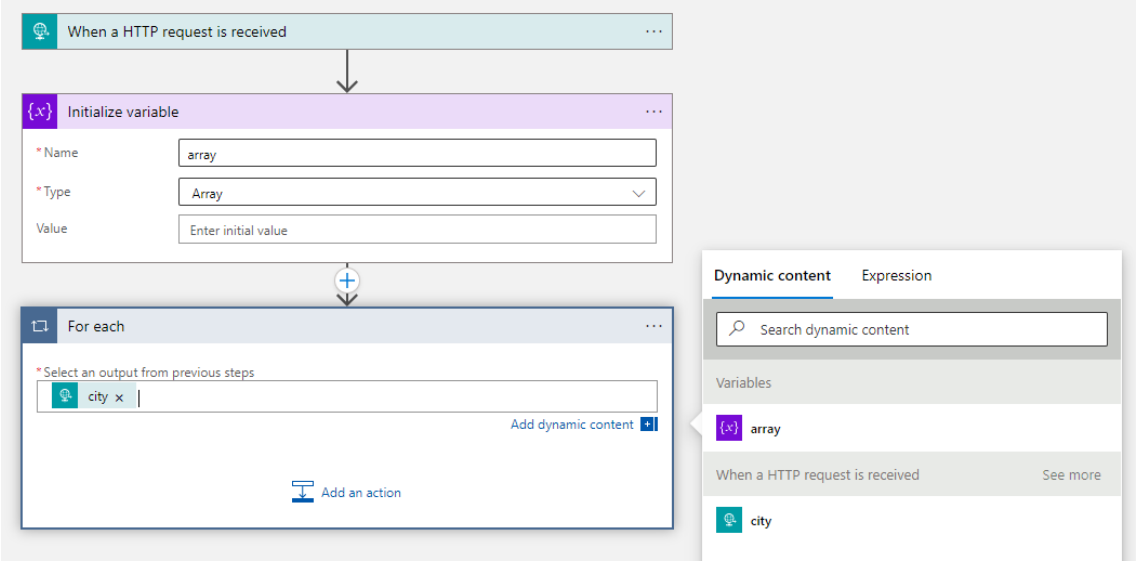
The trigger is given a sample payload from which to create a schema for the trigger.



Picture 7. Schema for the trigger.

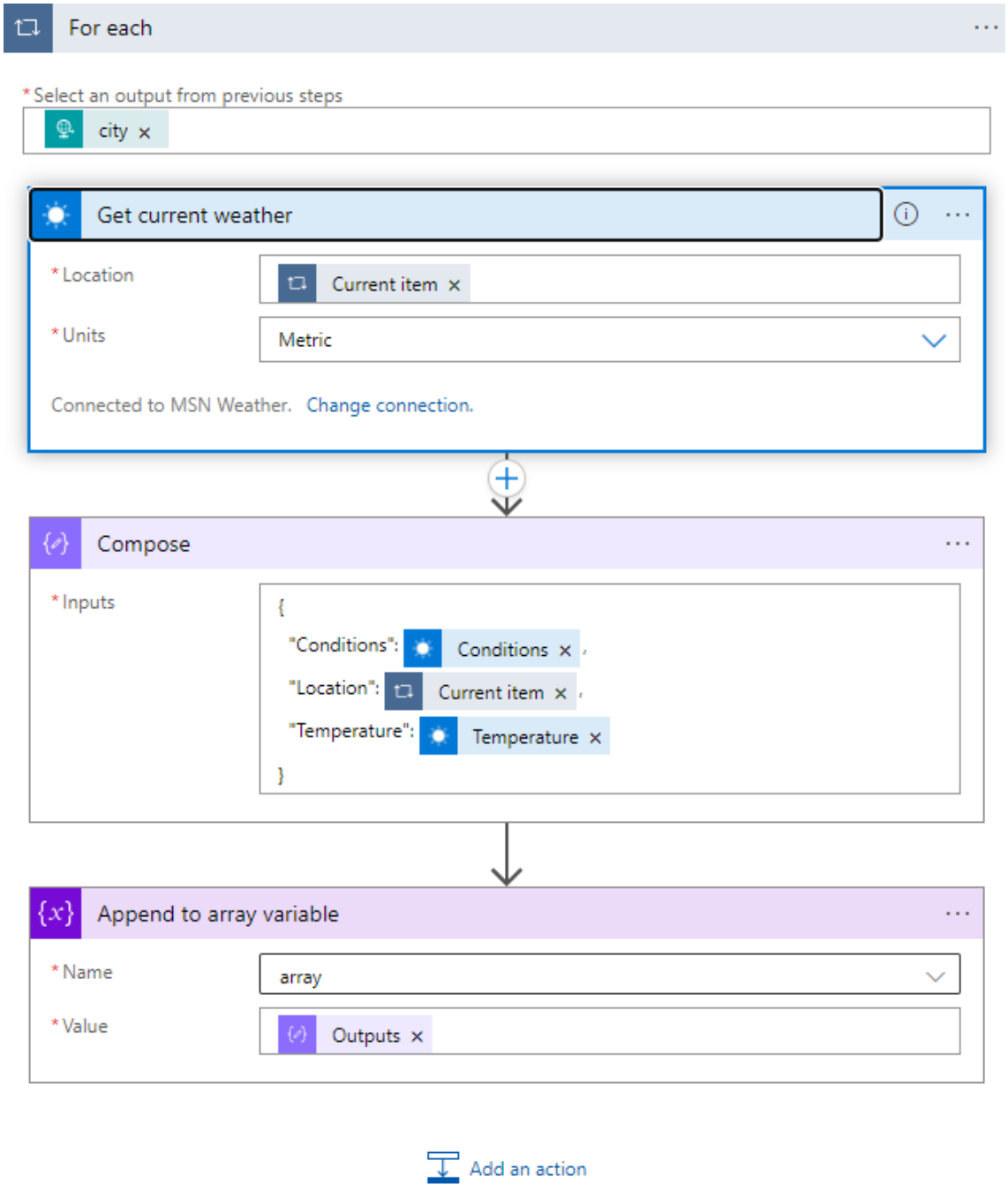
4.3 Adding actions to the Logic App

Now that the trigger is added and the Logic App is saved, a HTTP POST URL is generated. This is the URL that is invoked with Postman later. An array variable is initialized to which the data from MSN Weather API is stored later. After that a for each loop is created, so the weather data for all the cities contained in the payload can be retrieved. As input, “city” is selected from Dynamic content.



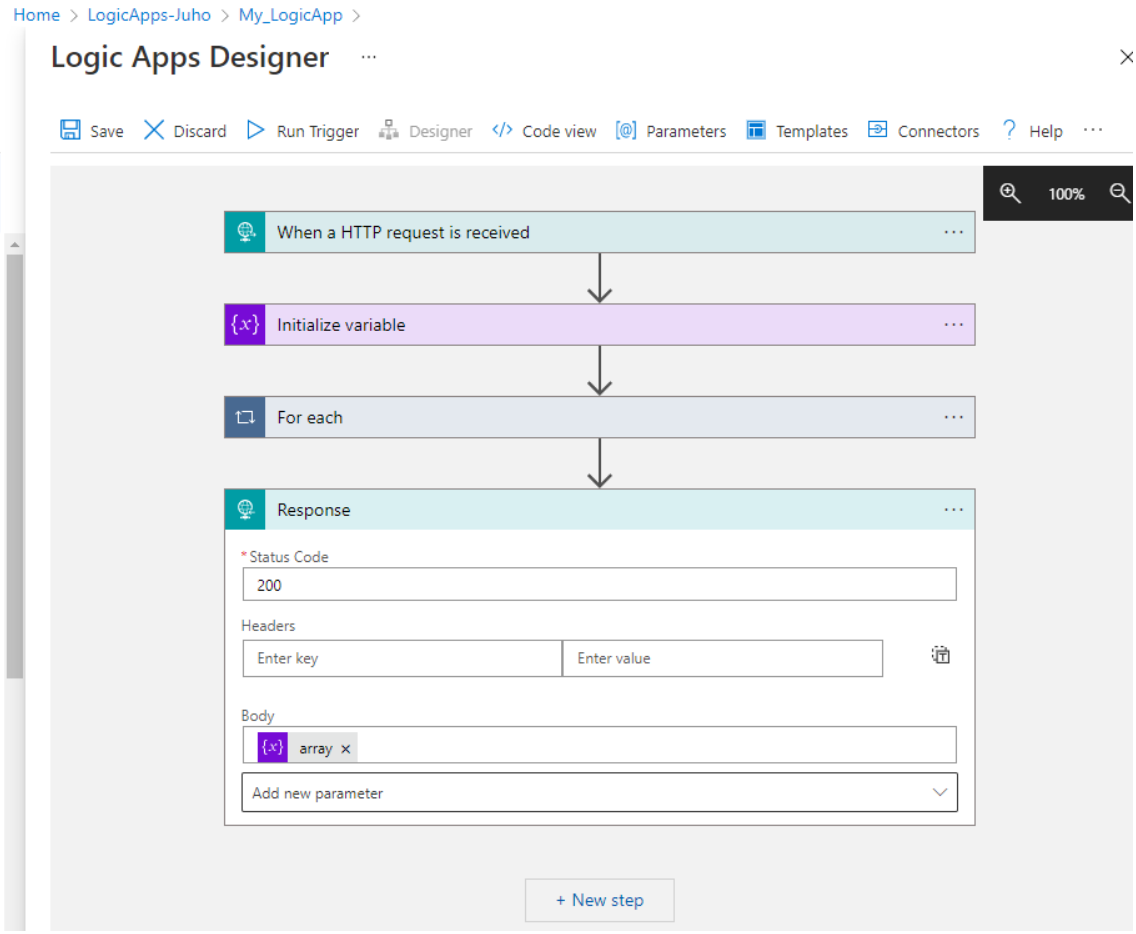
Picture 8. Initializing a variable and adding a for each loop.

Inside the for each loop, "Get current weather" action is selected from MSN Weather. "Current item" from dynamic content is selected as the location and units are set to metric. The "Get current weather" action contains a lot of data that is not wanted in this case, so a data operation is needed. Compose action is selected and custom JSON is written as inputs, containing only the name, conditions and temperature of each city. After that, the output of the Compose action is appended to the array variable created earlier.



Picture 9. Get current weather, data operation and append to array variable.

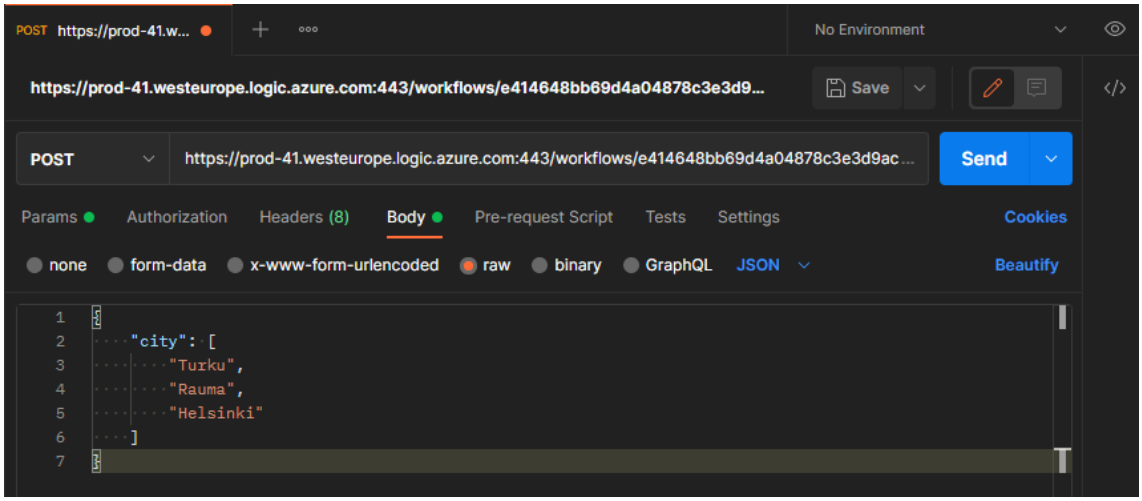
The For each loop is done. Last action will be response and it is created outside of the for each loop. As the response body, the array variable containing the weather data is selected. The Logic App is now ready to be invoked with Postman.



Picture 10. Ready Logic App.

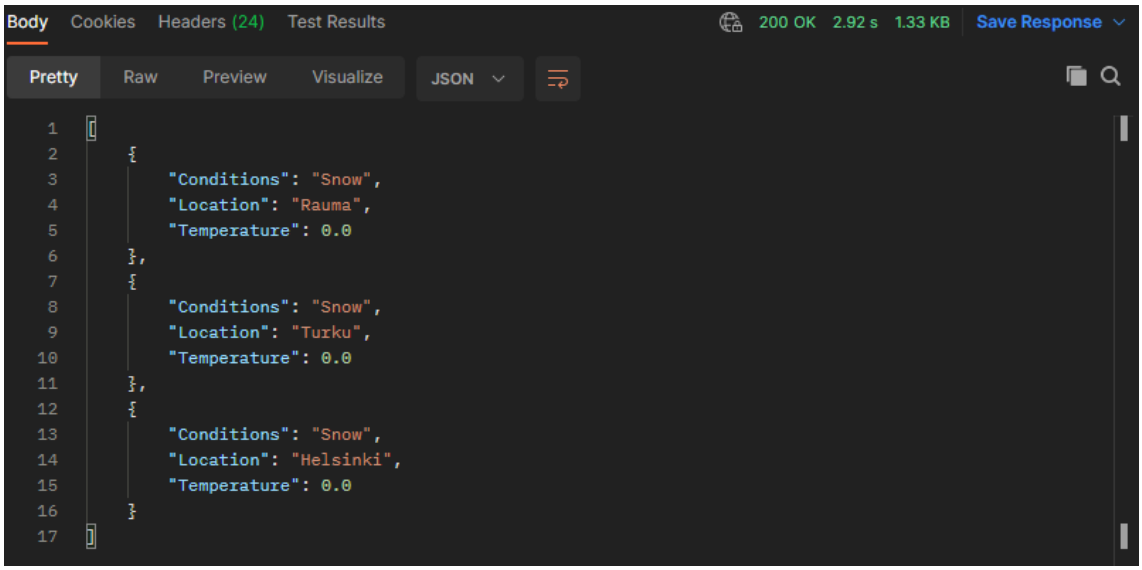
4.4 Invoking the Logic App with Postman

Next the Logic App is invoked using Postman. POST is selected as the request method and the HTTP POST URL from the trigger is copied as the address to which the request will be sent. Body is set as raw JSON containing wanted payload.



Picture 11. Postman with payload.

When "Send" is clicked, Postman will get status code 200 OK which means that the request was successful, and the response body contains wanted weather data for each city in a JSON array.



Picture 12. Response with weather data.

The Logic App is ready. The trigger was invoked to get weather data from MSN Weather API, the data was collected and parsed into an array and the array containing the transformed data was returned to the user as a response.

It would be easy to add actions and change actions to different ones to make the Logic App more diverse. The HTTP trigger could be changed to an Event Grid trigger that would trigger the Logic App every time the weather conditions change in either of the cities. Also, the response could be saved to Azure Blob Container for storing the weather data.

5 Conclusion

The goals of this thesis were to study Azure Integration Services and research their advantages compared to traditional on-premises integration solutions. All the main components of Azure Integration Services were researched and presented, and an example integration solution was created with Azure Logic Apps in Azure Portal.

The created Logic App was quite simple and serves as an example of how easy it is to build integration solutions with Microsoft Azure Integration Services. The Logic App has a basic structure with a trigger and actions. It could easily be developed further with for example changing the HTTP trigger into a event-based trigger and saving the response into a storage container.

Microsoft Azure Integration Services are a modern iPaaS solution for companies to create integration solutions between different systems and applications. With companies migrating their services, including integration services, increasingly to the public cloud from on-premises datacenters, Microsoft Azure is a choice to consider.

The components of Azure Integration Services work seamlessly with each other. With API Gateway taking in the request and Service Bus queuing and sending the messages to Event Grid that triggers a Logic App, everything is automated and requires minimal amount of upkeeping.

Integration solutions are and have been a critical part of businesses operations. Azure Integration Services provides a modern and easy to use alternative for creating integration services and solutions. Hence, it can be concluded that Azure Integration Services might be a solution for many more businesses in the future.

References

[1] Azure-Integrations-Services-Whitepaper-v1-0.pdf. Available: https://azure.microsoft.com/mediahandler/files/resourcefiles/azure-integration-services/Azure-Integration-Services-Whitepaper-v1-0.pdf [Accessed 15 January 2022].

[2] Introduction to Azure Fundamentals / What is Azure? Available: https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/what-is-microsoft-azure [Accessed 15 January 2022].

[3] Introduction to Azure Fundamentals / Tour of Azure services Available: https://docs.microsoft.com/en-us/learn/modules/intro-to-azure-fundamentals/tour-of-azure-services [Accessed 18 January 2022].

[4] Azure API Management Documentation / About API Management. Available: https://docs.microsoft.com/en-us/azure/api-management/api-management-key-concepts [Accessed 21 January 2022].

[5] Service Bus Messaging Documentation / What is Service Bus Messaging? Available: https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview [Accessed 21 January 2022].

[6] Azure Logic Apps Documentation / About Azure Logic Apps. Available: https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview [Accessed 03 February 2022].

[7] Azure Event Grid Documentation / What is Event Grid? Available: https://docs.microsoft.com/en-us/azure/event-grid/overview [Accessed 3 February 2022].

[8] Microsoft Azure portal. Available: https://azure.microsoft.com/en-us/feature2235s/azure-portal/ [Accessed 7 February 2022].

[9] Postman. Available: https://www.postman.com/ [Accessed 7 February 2022].

[10] Microsoft Power Platform and Azure Logic Apps connectors documentation / Connectors / Connector reference / MSN Weather. Available: https://docs.microsoft.com/en-us/connectors/msnweather/ [Accessed 18 February 2022].

[11] Azure Logic Apps documentation / Develop / Control workflow execution / Loops. Available: https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-control-flow-loops [Accessed 18 February 2022].

[12] Azure Logic Apps documentation / How-to guides / Perform data operations. Available: https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-perform-data-operations [Accessed 25 February 2022].

[13] Manage Azure resources documentation / How to / Manage resource groups / Azure portal. Available: https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/manage-resource-groups-portal [Accessed 25 February 2022].

[14] What is cloud computing? Available: https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#benefits [Accessed 3 March 2022].

[15] What is iPaaS? Guide to integration platform as a service. Available: https://www.techtarget.com/searchcloudcomputing/definition/iPaaS-integration-platform-as-a-service [Accessed 18 March 2022].

[16] Jaiswal, R. 2020. Azure Integration Services. Available: https://medium.com/@rakeshonrediff/azure-integration-services-438fe28d86c9 [Accessed 19 March 2022].