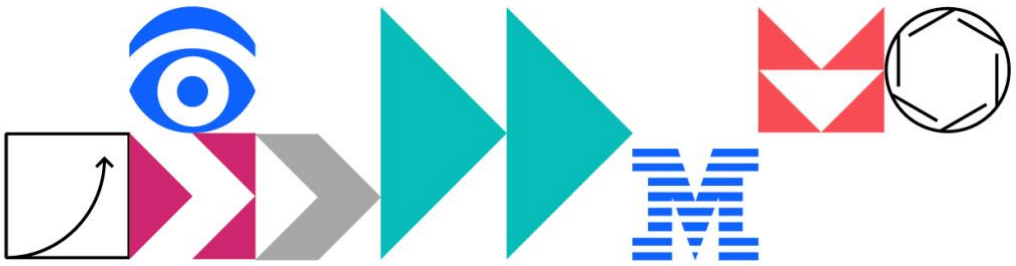
IBM TechXchange



Automation Decision Services

Session 4238

Lab Exercise Guide

Jorge D. Rodriguez

STSM | Business Automation SWAT

jorgedr@us.ibm.com

Swapnil Agrawal

Senior Product Manager – IBM Business Automation SaaS

aswapnil@ca.ibm.com

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.
Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information.

This document is distributed "as is" without any warranty, either express or implied. In no event, shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted per the terms and conditions of the agreements under which they are provided. The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled,
isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to ensure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer follows any law.

Notices and disclaimers (Continued)

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.** The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

Table of Contents

1 Introduction

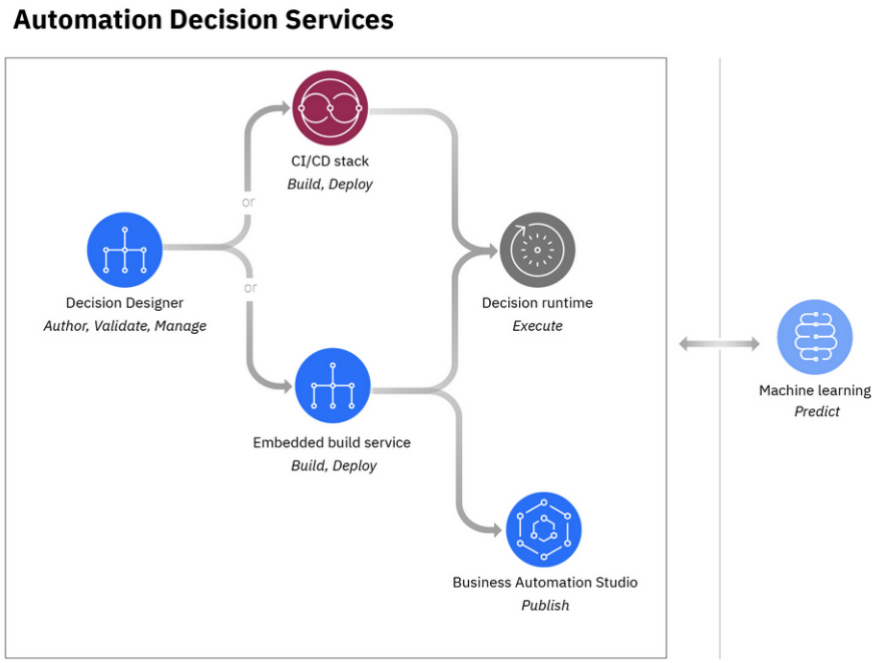
In this Lab, you will learn how to use IBM Automation Decision Services to create and manage operational decision services. You will learn key concepts of Automation Decision Services, such as modeling business decisions, implementing business rules, infusing decision services with predictive capabilities and decision service deployment.

1.1 IBM Automation Decision Services

IBM Automation Decision Services provides a comprehensive environment for authoring, managing, and running decision services.

Part of the IBM Cloud Pak® for Business Automation platform, Automation Decision Services, delivers advanced decision automation capabilities. Using Decision Designer in Business Automation Studio, business experts can model, author, and validate decisions in a low-code development environment. They can also infuse intelligence into business decisions by combining decision models and predictive models into unified decision services.

Automation Decision Services also integrates with a continuous integration and delivery (CI/CD) stack. You can build and deploy decision services directly from Decision Designer. Deployed decision services can then either be published as automation services in Business Automation Studio or invoked through the decision runtime as REST APIs.



For more information, see IBM Documentation
IBM Automation Decision Services
What is Automation Decision Services

1.2 Lab Overview

In this lab you will explore the capabilities of Automation Decision Services by extending a partially implemented decision service for a Client Onboarding scenario. You will have the opportunity to explore the use of data, decision and prediction models to create intelligent decision services using our low-code

authoring tool. You will add new business rules to the Client Onboarding decision service and will leverage predictive model capabilities to evaluate the client risk. Finally, you will deploy and test the changes made to the decision service.

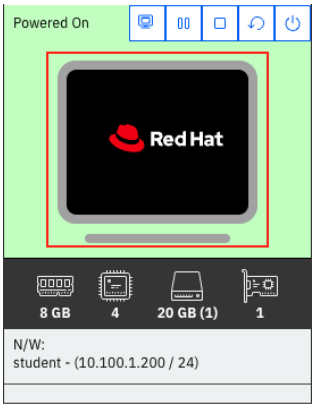
In this Lab you learn how to:

- Develop decision services using Decision Designer.
- Infuse intelligence into business decisions by adding a predictive model.
- Collaborate by sharing projects through a Git repository.
- Deploy decision services from Decision Designer.

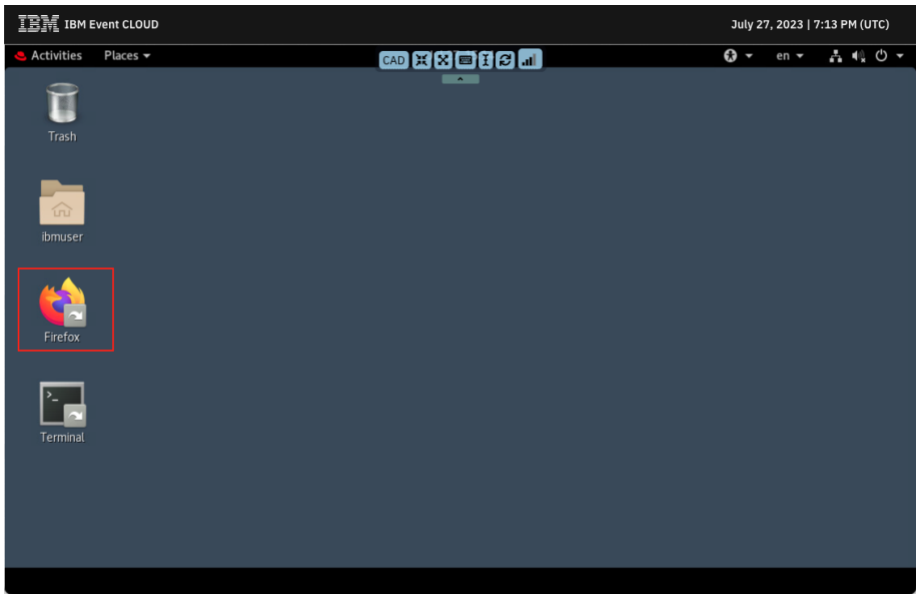
1.3 Lab Setup Instructions

All exercises in this lab will be completed in a remote virtual machine exclusively assigned to you for this lab session. You can access the remote virtual machine through the browser in your physical lab station. Once you have launched the lab environment from the physical lab station, follow the steps below to access your virtual machine:

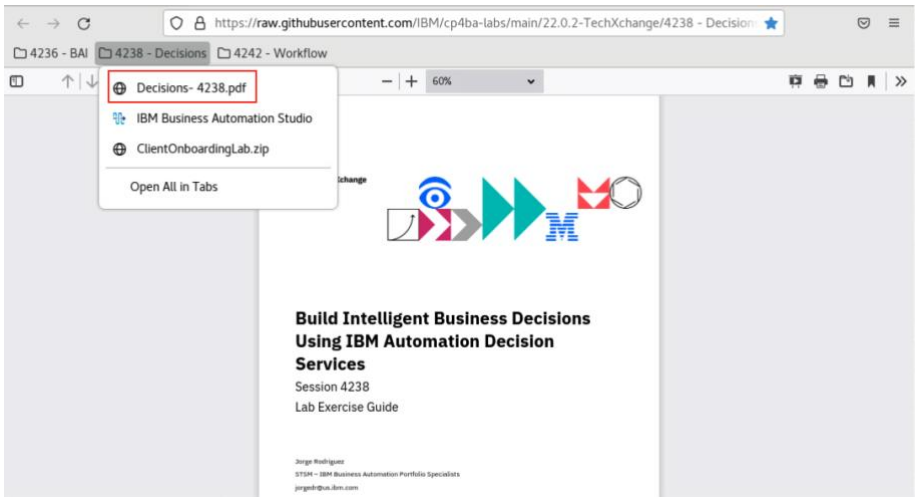
1. On the tile that represents your virtual machine, click over the monitor icon with the RedHat logo. This will open a new tab in your browser showing the remote virtual machine desktop.



2. Once in the virtual machine desktop, double click on top of the Firefox shortcut to open the Firefox browser in the remote virtual machine.

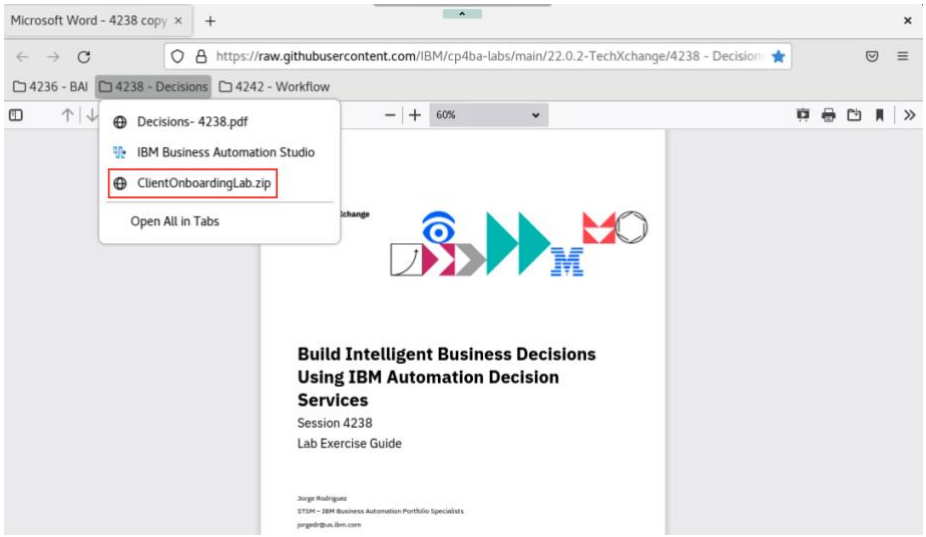


3. After Firefox opens, click over the **4238 - Decisions** folder found at the Bookmarks bar and click on the **Decisions - 4238.pdf** bookmark.

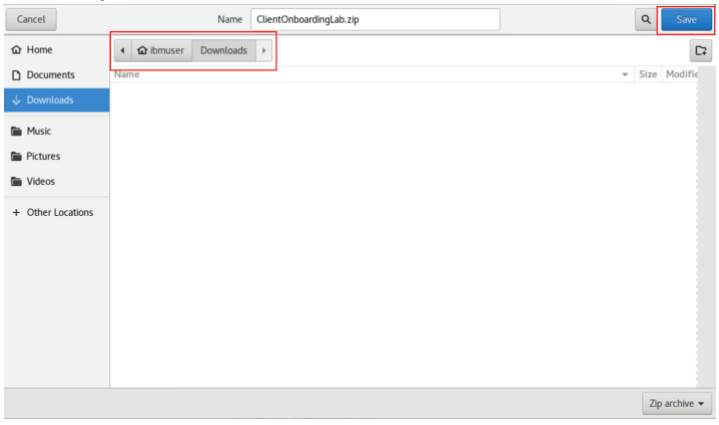


This will open a browser tab with a local copy of the lab instructions. While you are encouraged to follow the instructions available on your physical lab station, you can use this local version if needed. Keep this tab open for the duration of the lab.

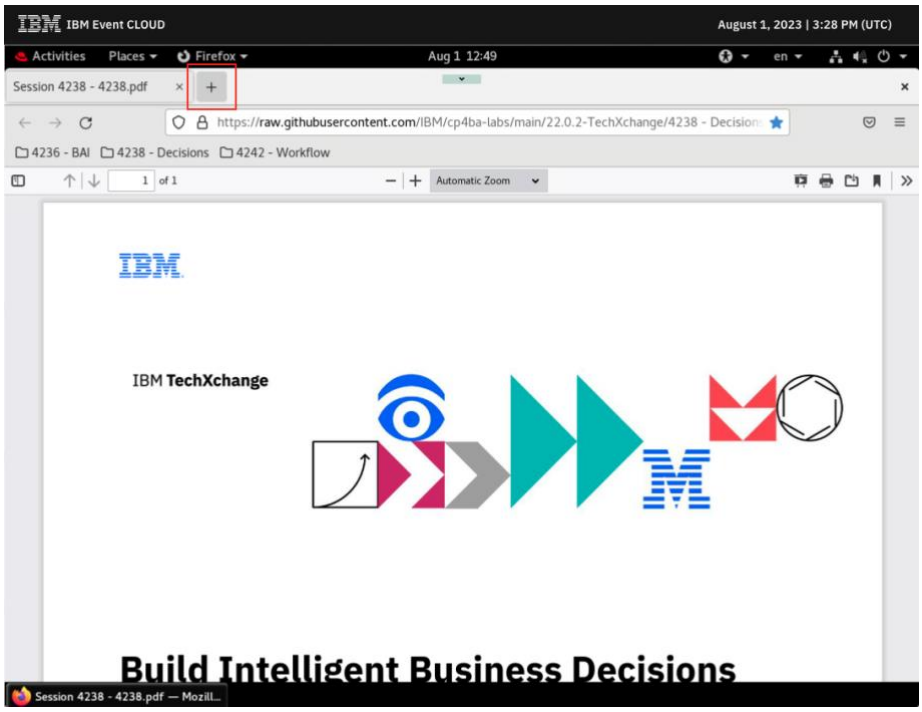
4. Now that you have opened a local copy of the lab instructions, you will download the **ClientOnboardingLab.zip** file that will be used later in this lab. To download the file, click on the **ClientOnboardingLab.zip** bookmark.



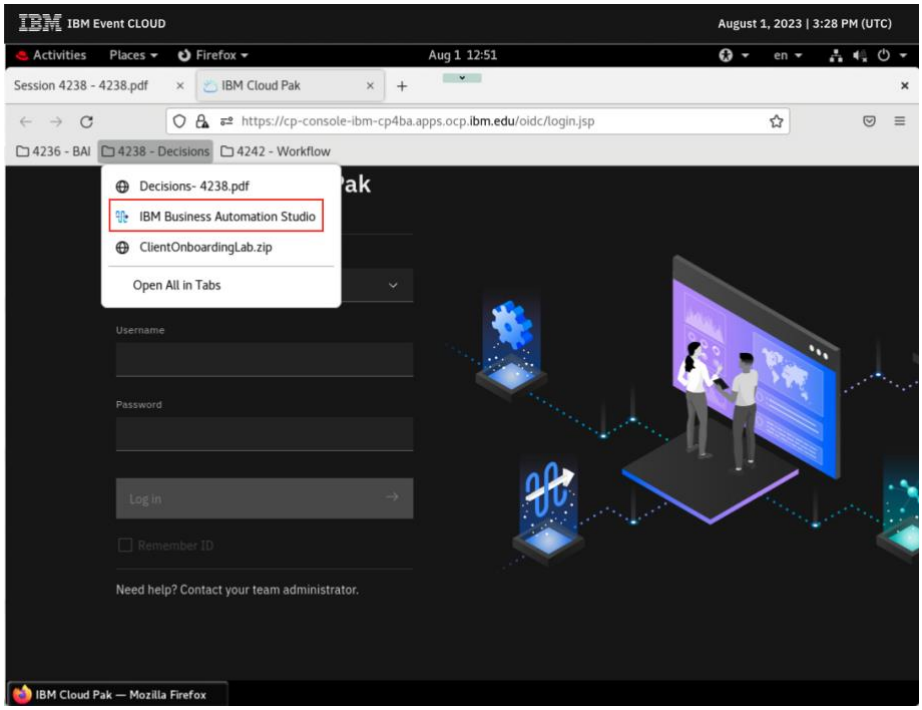
A dialog box will open. Click the **Save** button found on the top-right corner of the dialog box to save the file into your virtual machine. Notice the file will be downloaded into the **Downloads** directory.



5. Open a new browser tab by clicking the plus (**+**) sign found at the top part of your browser window.

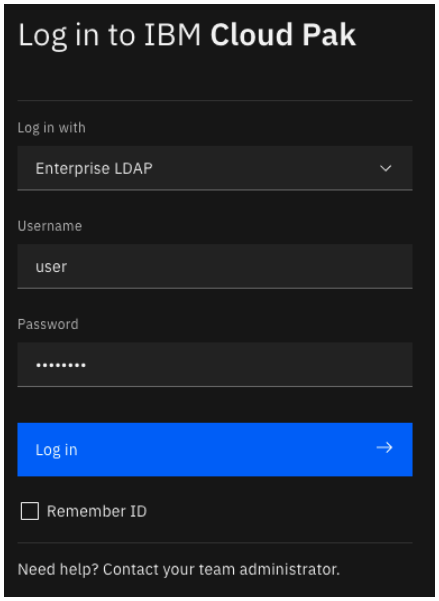


6. Click over the **4238 - Decisions** folder found at the Bookmarks bar and click on the **Business Automation Studio** bookmark. This link will take you to the **IBM Cloud Pak for Business Automation** Log in page. Keep this tab open for the duration of the lab. This will be your entry point to access the lab development environment.

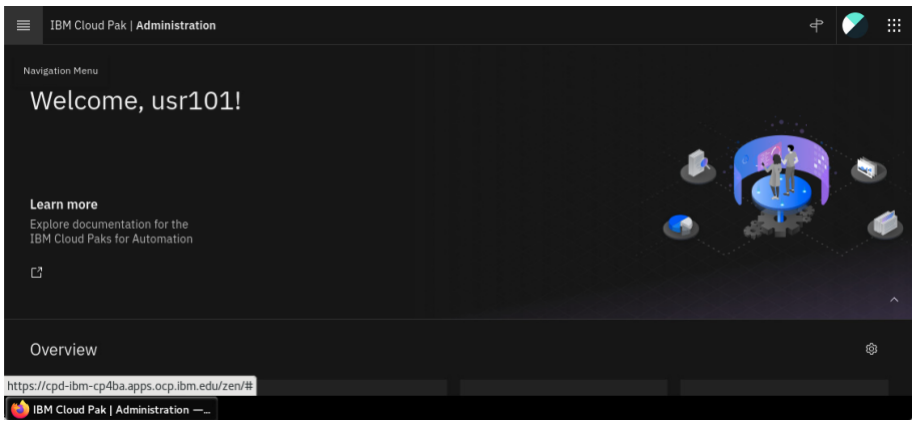


1.4 Log in to the Environment

1. On the login screen, under **Log in with**, select the **Enterprise LDAP** option.



2. Enter the username and password provided to you and click the **Log in** button. This will take you to the **IBM Cloud Pak for Business Automation** dashboard.



Congratulations you are now ready to start the lab exercises!

2 Exercise: Modeling Decisions

2.1 Introduction

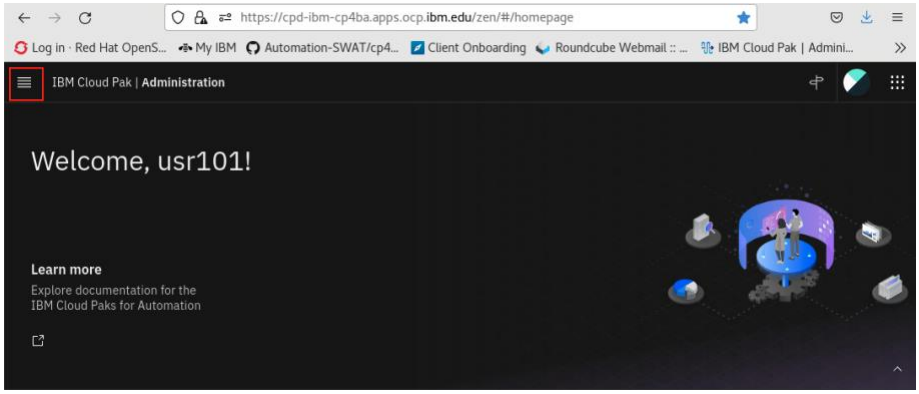
This exercise is an introduction to the authoring environment of Automation Decision Services. You will be using Decision Designer which is the development environment for creating decision services.
You will discover the main concepts of Automation Decision Services by exploring and modifying a partially implemented decision service. The scenario of this exercise is to calculate the fees for services the client is being onboarded to and to suggest additional services the client might be interested in.

2.2 Creating a Project and Importing a Decision Service

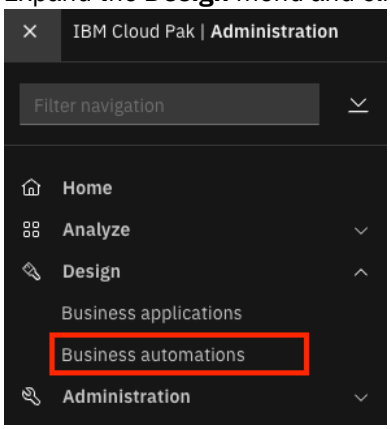
In this step, you will create a project in **Business Automation Studio** and will import a decision service from the zip file, **ClientOnboardingLab.zip**, prepared for this lab.

A project is a set of artifacts that share the same lifecycle and are grouped to solve a particular business problem. In this Lab you work with a decision automation project, it provides decision capabilities to help business experts model and automate repeatable decisions.

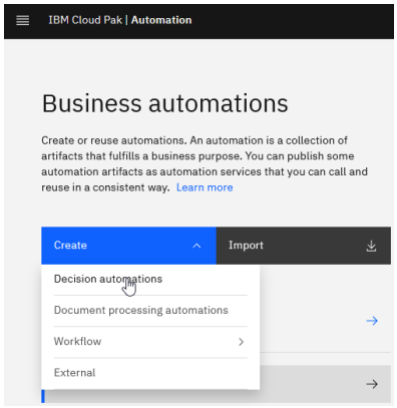
1. From the **IBM Cloud Pak for Business Automation** dashboard, click on the hamburger menu icon found at the top-left corner.



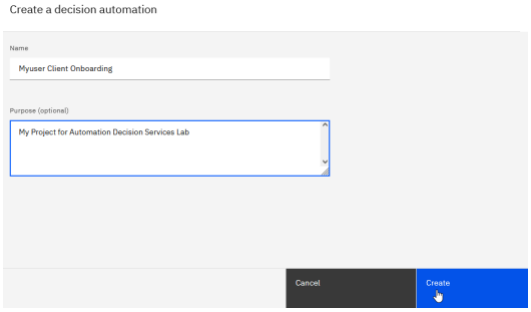
2. Expand the **Design** menu and click on **Business automations**.



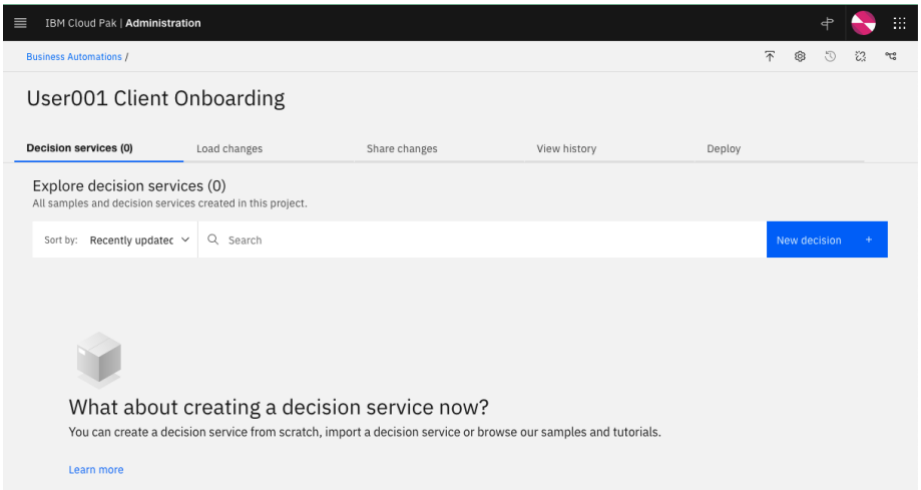
3. Click **Create** and select **Decision automations**.



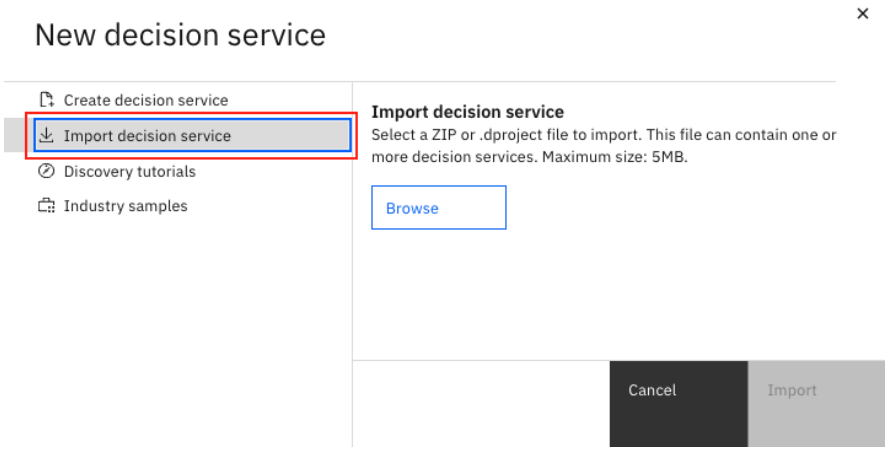
4. In the **Name** field, enter **UsrNNN Client Onboarding Decision** where *UsrNNN* is your assigned username, then click **Create**.



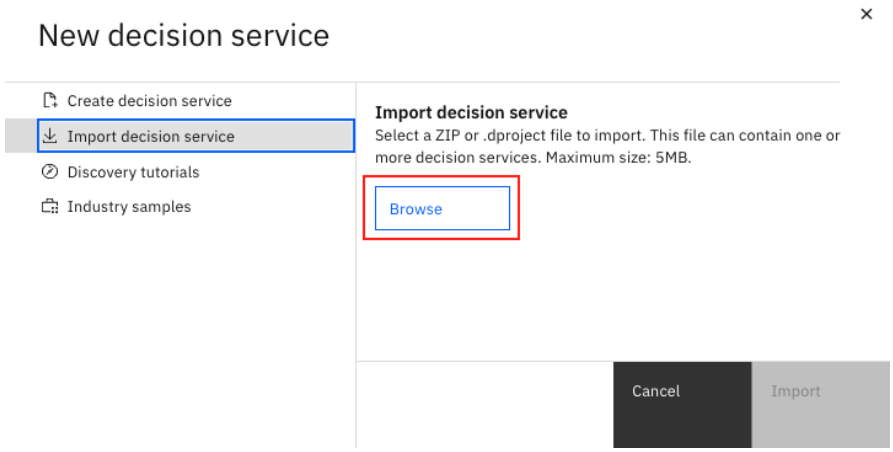
5. Your new project opens in Decision Designer. Do not follow the guided popup, if it shows up click **Maybe Later** to continue.
6. Click the **New decision +** button.



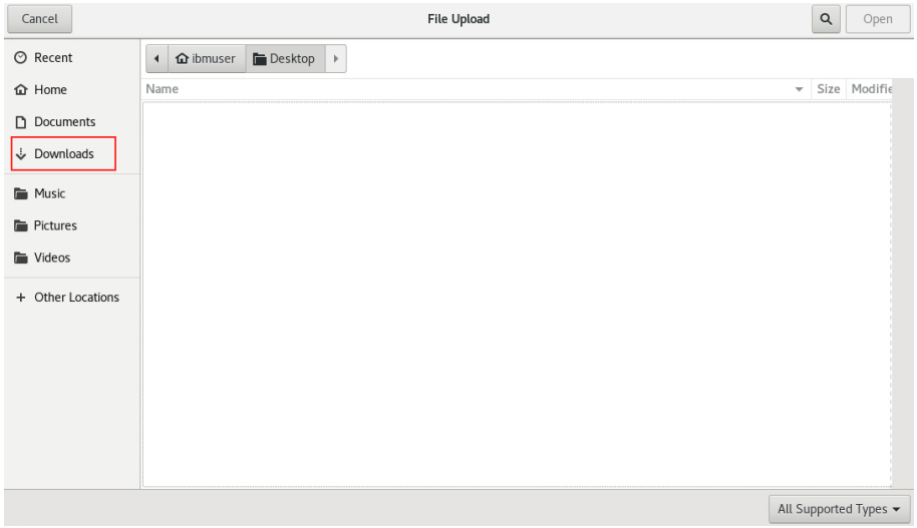
7. Select the **Import decision service** section on the left to import the decision service provided.



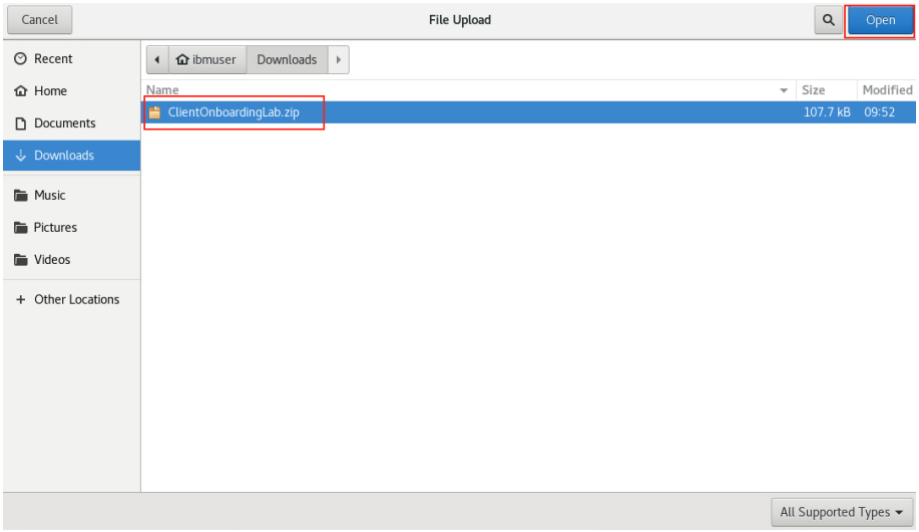
8. Click **Browse** to select the file prepared for this Lab, **ClientOnboardingLab.zip**. You previously downloaded this file while following the Lab Setup Instructions section.



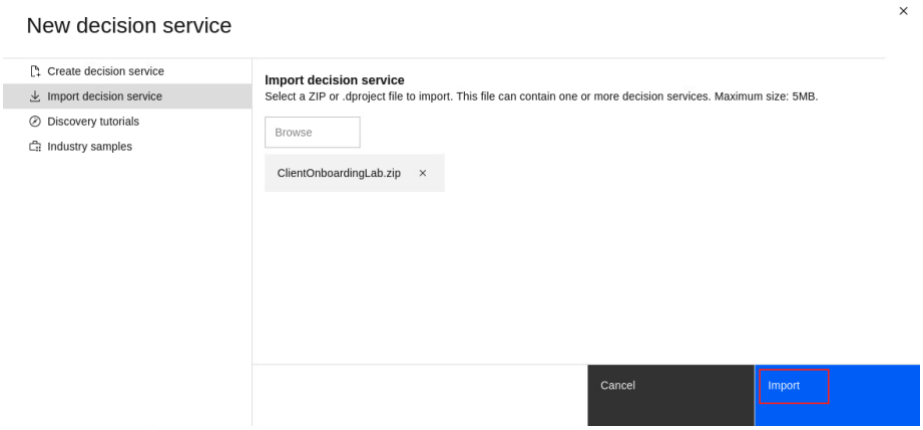
On the dialog box select the Downloads directory by clicking the **Downloads** option to the right.



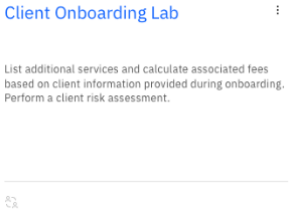
Make sure the **ClientOnboardingLab.zip** file is selected and click the **Open** button on the upper-right.



9. Back on the **New decision service** window. Click **Import**.



10. A tile of a decision service named **Client Onboarding Lab** will show up on the project page.



Once you see the **Client Onboarding Lab** tile is in your project, you have successfully imported the decision service used for the lab.

2.3 Exploring the Decision Service

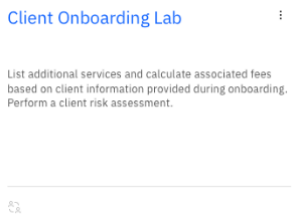
In this step, you explore the decision service imported in the previous section. A decision service uses the following decision artifacts to define business decisions:

- **Decision models**: Represent a diagram that expresses the business logic. You can reuse the output of a decision model in another decision model.
- **Predictive models**: Apply data from a machine learning model to make a prediction.
- **Data model**: Represents the data structure used by the business logic. You can use the data model vocabulary in your decision models and predictive models.
- **External libraries and data sources**: Extend the data models using Java code that contain data models and functions that you can use in decision models (not covered in this lab).

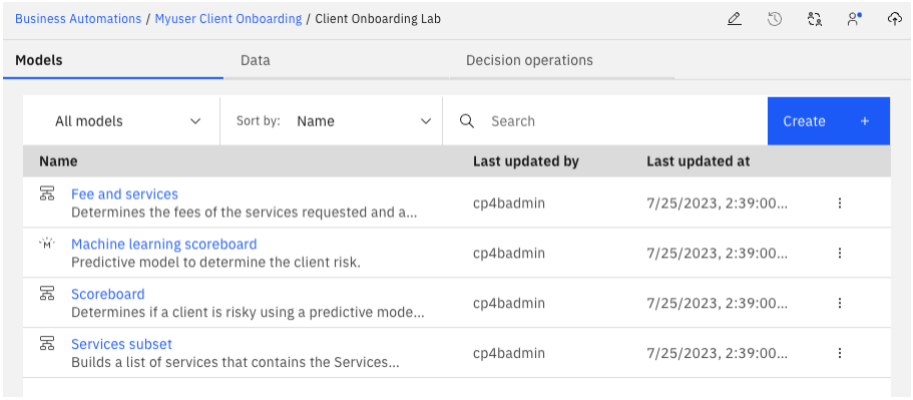
- **Decision operations**: Define entry points for decision services execution.
- **Task models**: Allow business experts to define the decision logic outside model diagram (Note: task models are not covered in this lab).

For more information, see IBM Documentation Building decision models.

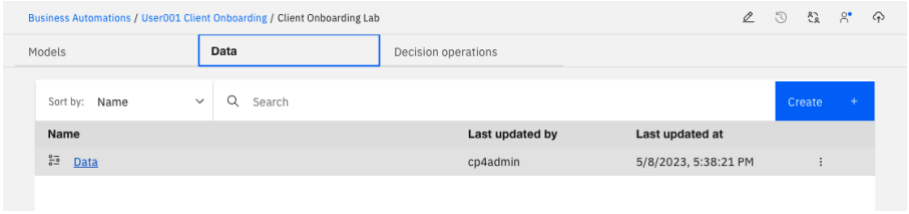
1. Click on the **Client Onboarding Lab** tile to open the **Client Onboarding** decision service.



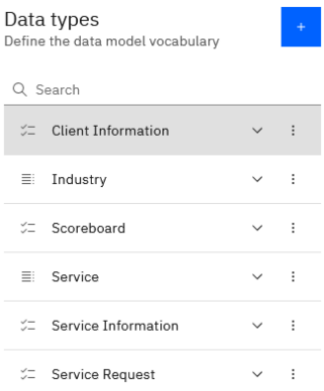
2. The decision service and the artifacts created are displayed. You will explore the data model first.



3. Open the **Data** tab and click on **Data**.



A data model defines the data that is needed to make your decisions. In a data model, you use predefined and custom data types to create a vocabulary that you can use to populate decisions. Basic data types, including string and integer are predefined by default in Automation Decision Services. You create a data model to define custom data types to match specific needs for your service. Data models can be used across decision models in a service.



The data model for Client Onboarding contains 6 Data types: **Client Information**, **Industry**, **Scoreboard**, **Service**, **Service Information**, and **Service Request**.

Two categories of data types can be defined:

- The **enumeration type**, such as Industry and Service, which contain a list of possible values.
- The **composite type**, such as Client Information, Service Information, and Service Request, which contain a set of attributes with a name and a type. The type can be a basic type predefined in Automation Decision Services such as string or integer or a custom data type.

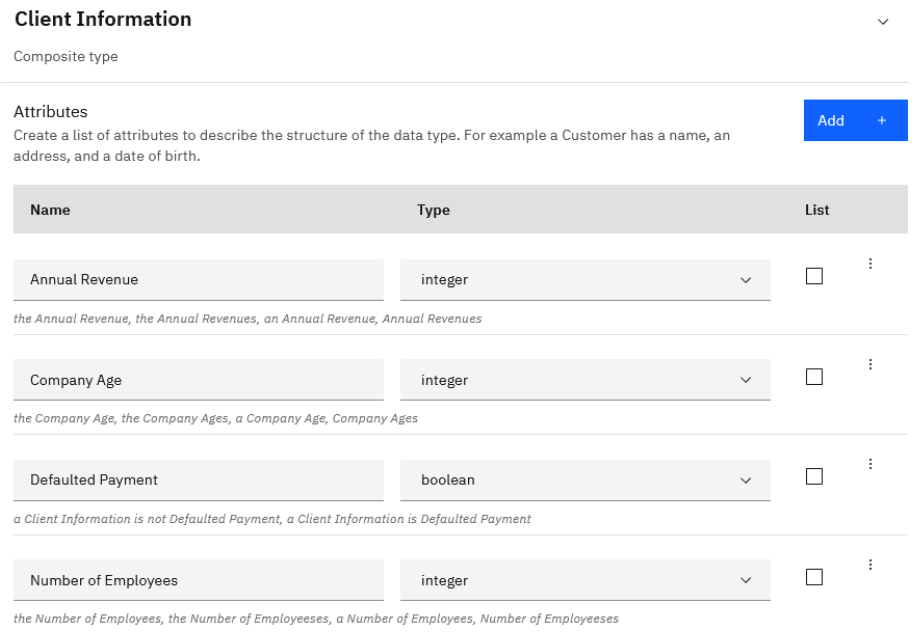
The types defined as input of the model for this exercise (**Fee and services**) are:

- The enumeration **Industry**: lists the available industries.
- The enumeration **Service**: lists the available services.
- The composite type **Service request**: describes in which industry the client is and what are the services requested.
- The composite type **Client Information**: describes the characteristics of the client including annual revenue, number of employees, company age, defaulted payment.

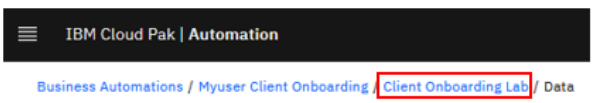
The type defined as output of the model for this exercise (Fee and services) is:

- **Service Information**: a list of additional services suggested to the client and the fee for the services requested.

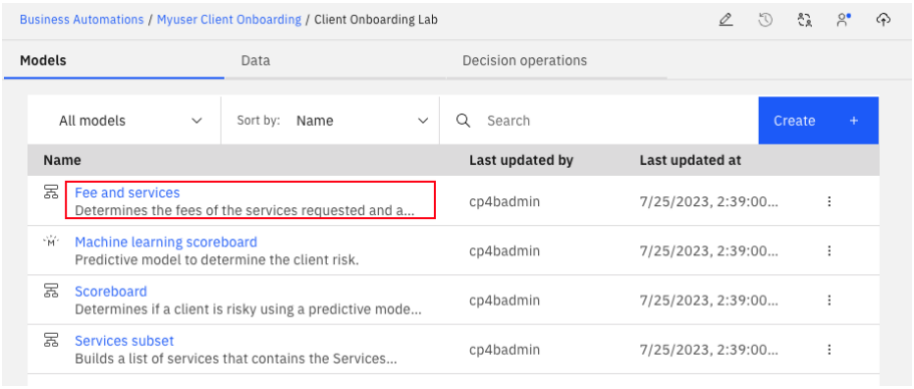
4. Click on **Client Information** to explore the data type. It is a composite type with four attributes. Attributes and values allow you to define the characteristics of a data type.



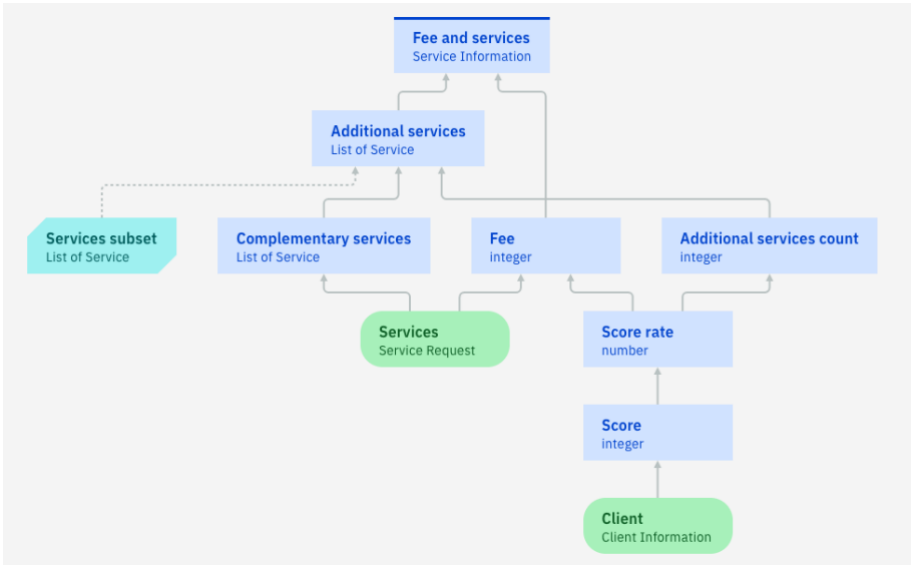
5. Click on the **Client Onboarding Lab** breadcrumb to navigate back to your decision service. You will now explore the decision model.



6. In the **Models** tab, click on **Fee and services**.

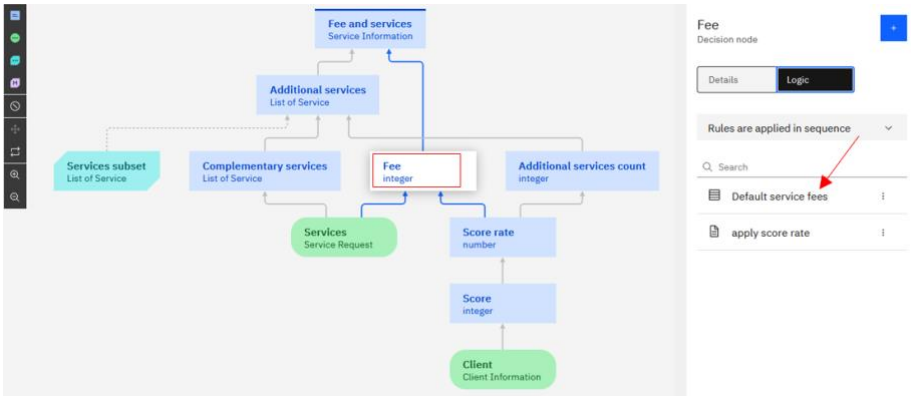


In Automation Decision Services, you implement your decision by building a decision model diagram. Once on the **Fee and services** pane you will see the diagram for this model. The diagram is composed of nodes and links. The decision nodes, in blue, contain logic that defines how each decision is made. Input nodes, in green, represent the data used to determine the decision output. Function nodes, in cyan, represent values that are computed from other decision models. The links represent the relationship between the nodes and show the input that is available in a node.



In this exercise, the top decision node **Fee and services** represents the final decision. It calculates the fees for the services requested by the client and suggests additional services to offer. It is based on the **Client** information and the **Services** requested as input. To provide the outcome, it depends on intermediate steps, the sub-decisions **Fee** and **Additional services.**

7. Click on the **Fee** node, and then select the **Logic** tab to explore the decision logic. It displays the decision table **Default service fees** and the rule **apply score rate**. Both of these rules implement the business logic of the fee calculation. In Automation Decision Services, you express decision logic with a set of business rules and/or decision tables.



8. Click on **Default service fees** in the **Logic** tab to open the decision table.

Default service fees

Edit preconditions

	Services count	Fees
1	0	0
2	1	15,000
3	2	25,000
4	3	38,000
5	4	50,000
6		

Each row represents a single rule where the **Services count** column represents a condition parameter and the **Fees** column the value of the action when the condition is met.

9. Click on **apply score rate** in the **Logic** tab to open the business rule.

apply score rate

Type your rule using the list below as reference

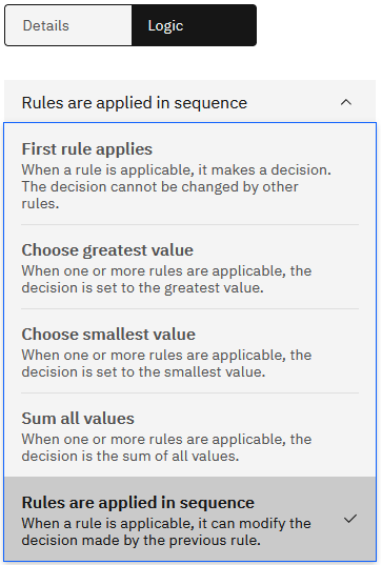
```
1
2    set decision to round ( decision * 'Score rate', 0 );
```

Inputs (2) Output (1)

Services	Service Request
Score rate	number

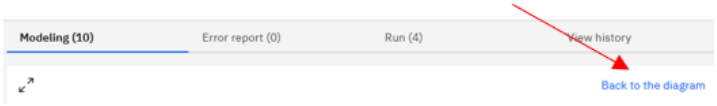
The **apply score rate** is a simple business rule statement that determines a rate to apply to the fee calculation.

10. On the Logic tab, expand **Rules are applied in sequence** drop down. This widget allows you to select the **rule interaction policy**. The rule interaction policy defines how the rules defined in the node interact with each other. In this decision node, the rules are applied in sequence. A fee is assigned according to the number of services in the **Default service fees** rule. Then, a rate is applied to calculate the final fee for the requested services.



The interaction policy choices differ according to decision node type (number, list, integer...). In this exercise, the decision node **Fee** has an output type of integer, so the options above are displayed for the integer type.
For more information, see IBM Documentation Choosing an interaction policy.

11. Click on **Back to the diagram** to return to the decision diagram.

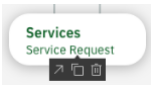


2.4 Updating Decision Logic

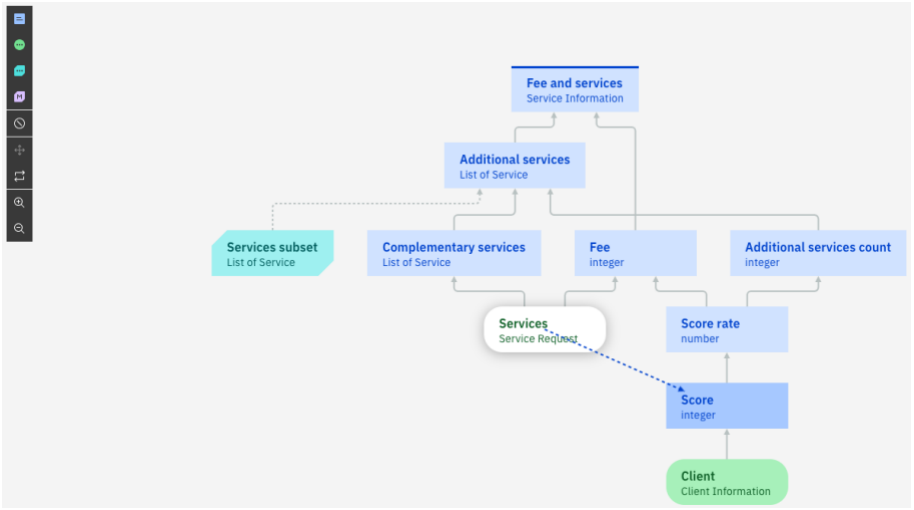
You will now change the decision logic for the **Score** node to consider the number of services requested by the client. A higher number of services should lead to a higher client score.

The change in the **Score** decision logic implies:

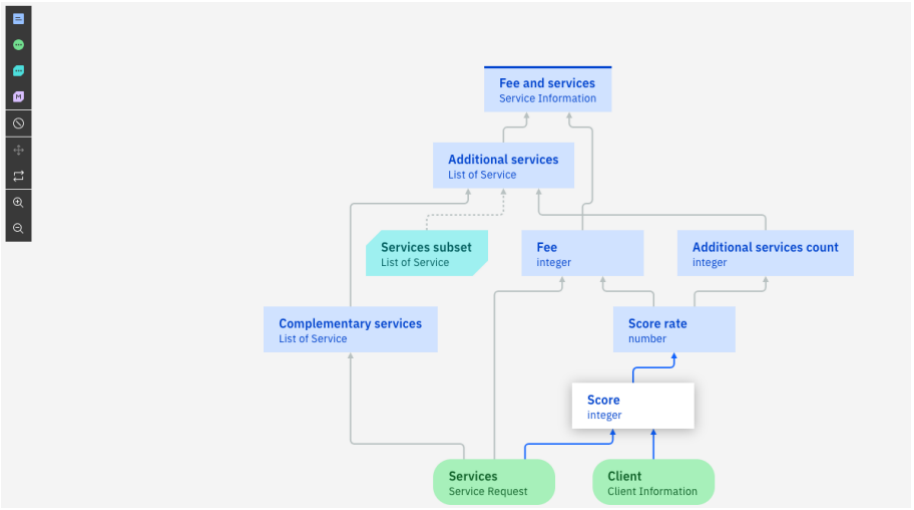
- Knowing the number of requested services from the **Score** node. That requires adding a link from the **Services** input node to the **Score** node.
- Adding the computation based on the number of services. That requires adding a rule to compute the new score.

1. Click on the **Services** input node, then hover over the icons 

. Single click on **Connect to another node** (arrow icon) to add a link to connect to another node. Move your mouse to the **Score** decision node and click again.

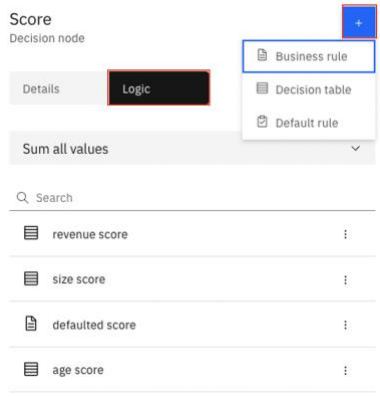


A link is created and the decision diagram is updated to reflect the new dependency.



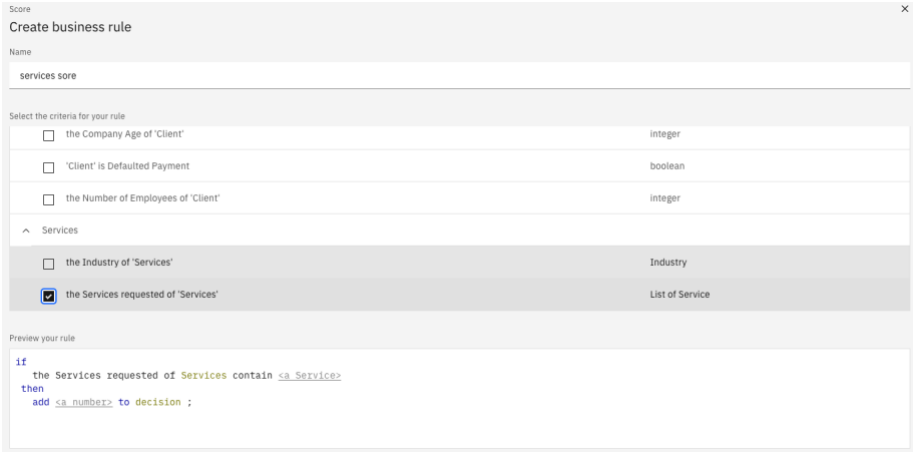
2. Click on the **Score** decision node, then select the **Logic** tab on the right pane. Currently the client score is calculated using four different rules. Notice the **rule interaction policy** is set to **Sum all values**. This means the output of the **Score** node will be calculated adding up the outcome of each individual rule.

   Click on the plus icon () and click **Business rule** to add one more rule.



A wizard opens to help you select the criteria of your rule.

3. Enter **services score** as the name of the rule, and then scroll down to select the criteria **the Services requested of 'Services'**. The list of criteria depends on the inputs of the decision node. Selecting a criteria provides you with a more complete draft of the rule when it is first created, but you can edit it entirely after the rule is created.

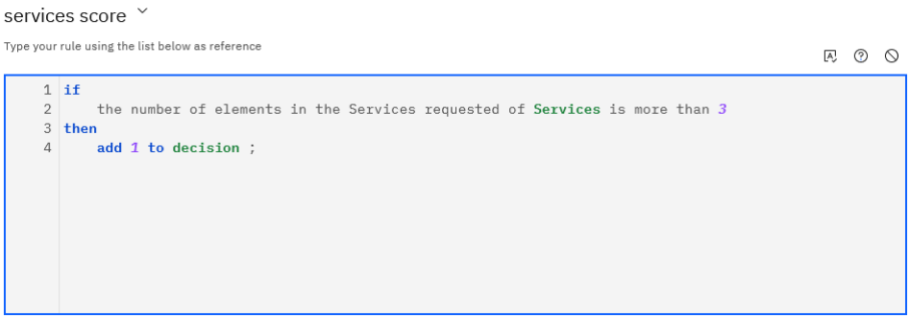


4. Click on **Create**. You see a red icon that indicates an error. You can hover over the icon to see the details.

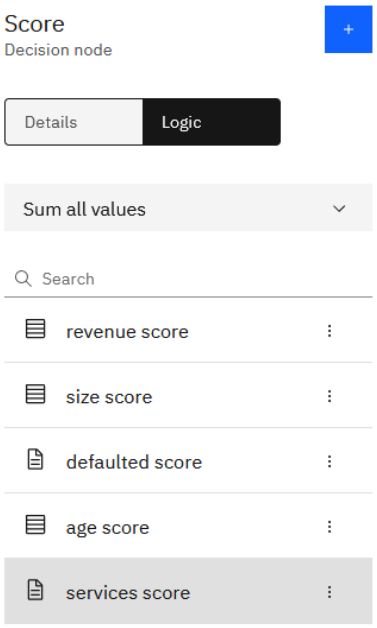
5. In the rule editor, replace the existing business rule statements with the text below. Use copy and paste to avoid errors.

```
if
    the number of elements in the Services requested of Services is more than 3
then
    add 1 to decision ;
```

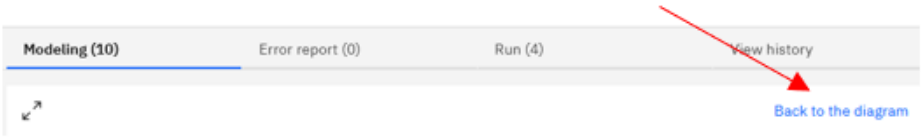
Once finished, the rule appears as follows:



The decision logic in the **Score** node now includes a fifth rule that takes into consideration the number of services requested by the client. If the number of services requested are above three, the client score is increased.



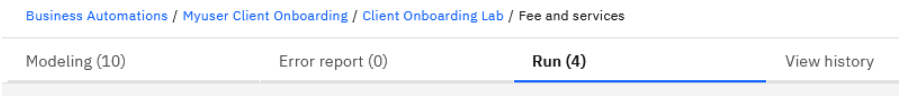
6. Click on **Back to the diagram** to return the decision model diagram.



2.5 Test the Decision Service

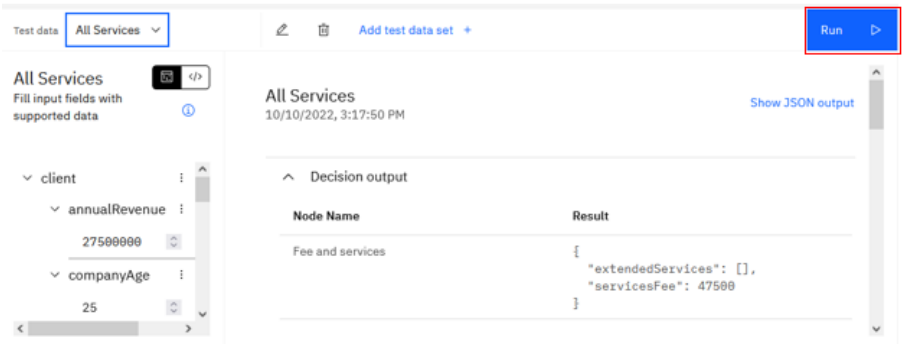
Now that you have modified the decision model and updated the decision logic, you will want to test the decision service before deploying or making further changes. Automation Decision Services lets you test and fine tune the models you build directly in Decision Designer.

1. Select the **Run** tab.



   On the left side of the screen, the **Test data** pane lets you select the test data to submit to execute the rules. In this Lab, four test data sets have been pre-defined for you: **All Services**, **Federal 2**, **Federal 3**, **Telecom 1**.

2. Select **All Services** and click on **Run**.

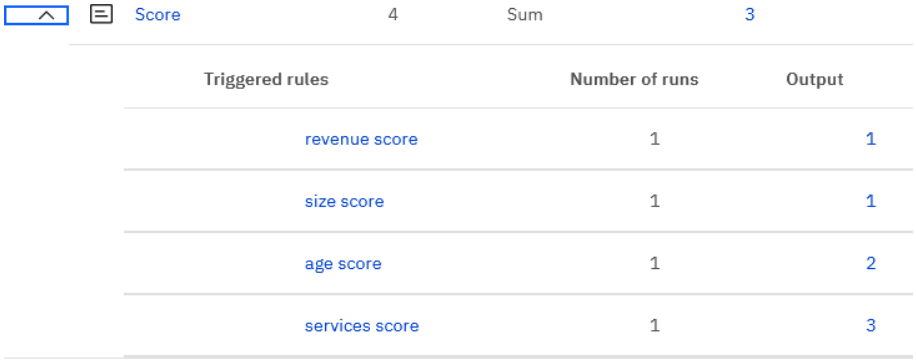


   A report is generated which includes the input and output of each node in the decision model. The final decision is displayed at the top of the report: for Fee and services the result is `"servicesFee":`

47500.

Expand **Run history**, and then expand **Score**.
It shows that the rule you added, **services score**, has been triggered along with the rest of the pre-existing rules. The result for the decision node **Score** is a consolidated result. It is the sum of the rules that triggered the result, applied in sequence.

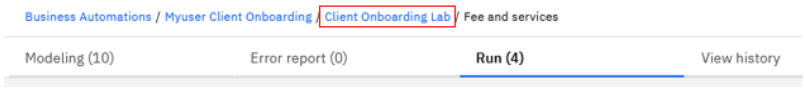
Each displayed **Output** is the sum, including the previous output.



In this example, the output of the **Score** decision is 3:

- revenue score output is 1
- then added to **size score** (size score is 0) it remains 1
- then added to **age score** (age score is 1) the result is 2
- then added to **services score** (services score is 1) the final result is 3

3. Click on the **Client Onboarding Lab** breadcrumb to navigate back to the **Models** tab.



2.6 Summary

You have completed the Modeling Decisions exercise. You created a project and a decision service to define the fee and services for a client being onboarded to a set of services.

3 Exercise: Adding Machine Learning to Decision Model

3.1 Introduction

In this exercise you learn about how to use predictive models in Automation Decision Services to infuse your business decisions with predictive insights. Adding a predictive model to the **Client Onboarding** decision service allows you to benefit from historical data and the experience of previous customers to estimate the client's risk level.

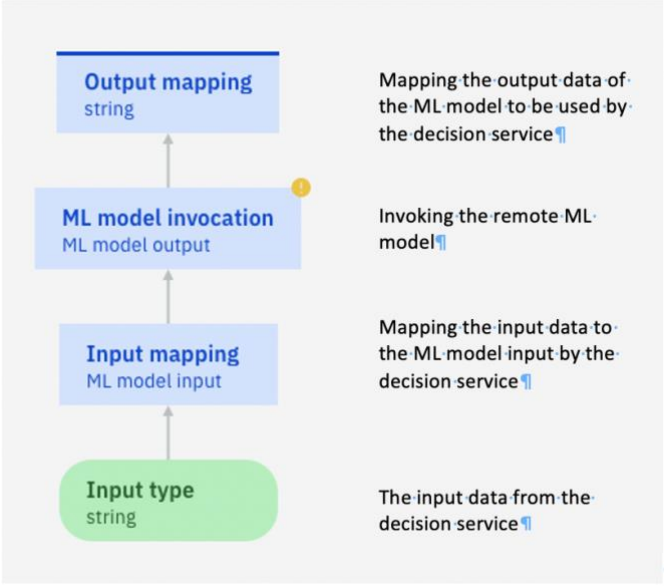
A **predictive model** is a wrapper used in Automation Decision Services to encapsulate and call a machine learning model. It allows you to map the inputs and outputs of the machine learning model into the data model of your decision service. It also allows you to embed the predictive capabilities of the machine learning model into other decision models in your service.

In this lab, you will use a predictive model and a machine learning model created to complete this exercise. The machine learning model evaluates client risk based on information on the client such as annual revenue, company age, number of employees, and industry and provides a prediction of the client risk.

For more information on predictive models see the IBM Knowledge center documentation, Creating a predictive model.

3.2 Exploring the Predictive Model

The structure of a predictive model is defined through a model diagram. The diagram is similar to a simple decision model diagram. When a predictive model is first created the following elements are generated for you:



Once you create a predictive model, you must connect it to a **machine learning model**. Machine learning models are created by data scientists using historical data and deployed on a machine learning platform, such as Watson™ Machine. In Decision Designer, these types of machine learning models are referred to as **remote machine learning models**. To use a remote machine learning model from Automation Decision Services you must configure the access to the machine learning provider that contains the model deployment. Alternatively, data scientists can provide you with a **transparent machine learning model** that can be imported as a simple `.xml` or `.pmml` file into Decision Designer without any additional configuration of the platform. This type of machine learning model is referred to as a **local machine learning model**. The

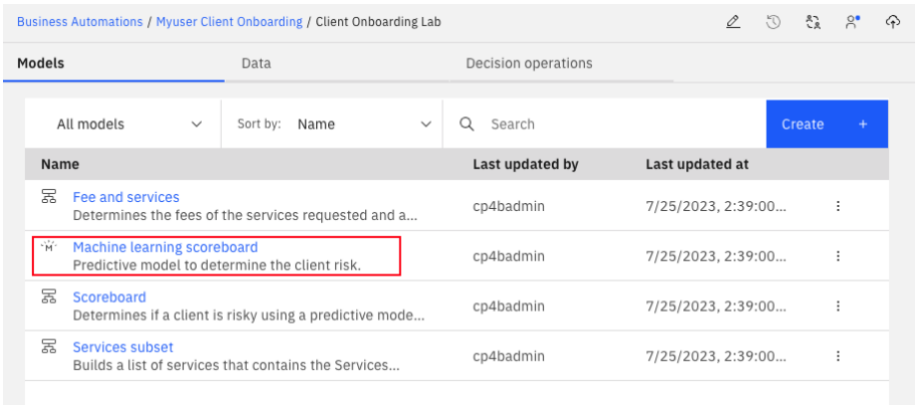
predictive model created for this Lab leverages a **local machine learning model** imported from a file named **clientDefaultedPaymentRS.xml**. You will explore this configuration later in this section.

For more information about how to use machine learning models in Automation Decision Services see Integrating machine learning in the IBM Knowledge Center.

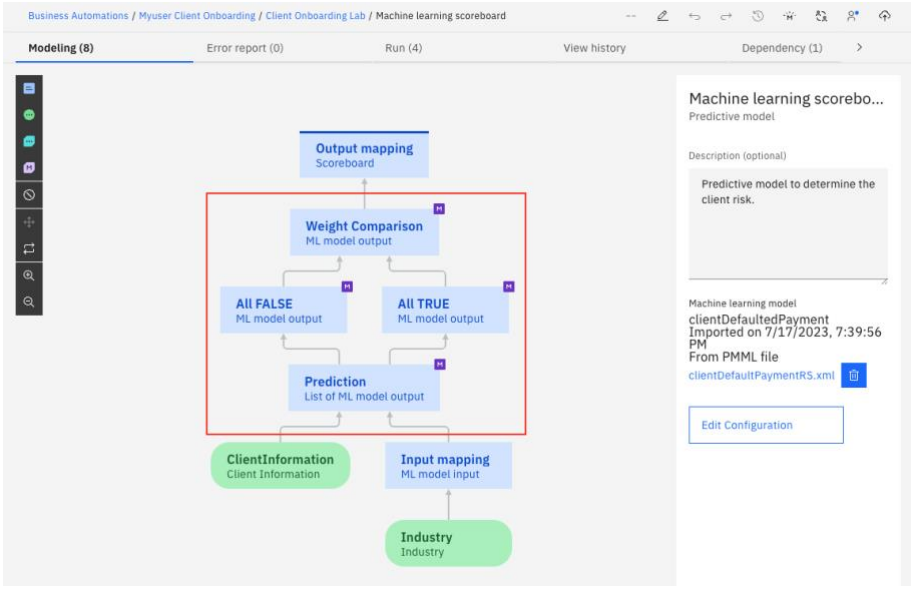
When connecting the machine learning model to the predictive model, you must map the input and output data elements used by the machine learning model to the data types and variable names of the data model used in the decision service. This is required so that the right values and data types are used when calling the machine learning model and to ensure the outcome of the predictions are stored in your decision data model properly. While most of the data mapping can be done when you first configure the predictive model, some values require more complex computations. This can be implemented as business rules or decision tables in the **Input mapping** and **Output mapping** nodes of the predictive model.

You can explore some of the concepts explained in this section by looking at the predictive model provided in this Lab.

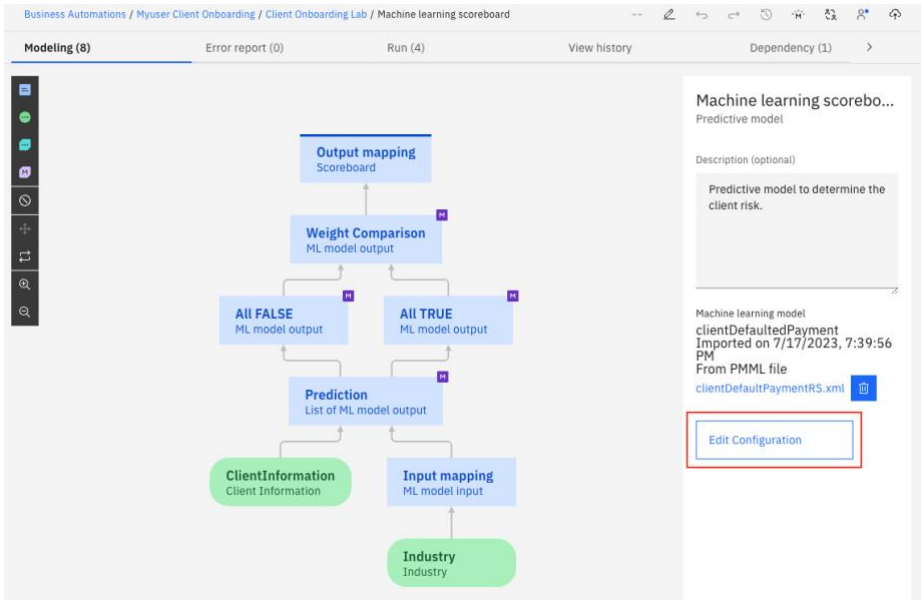
1. Click on **Machine learning scoreboard** under the **Models** tab.



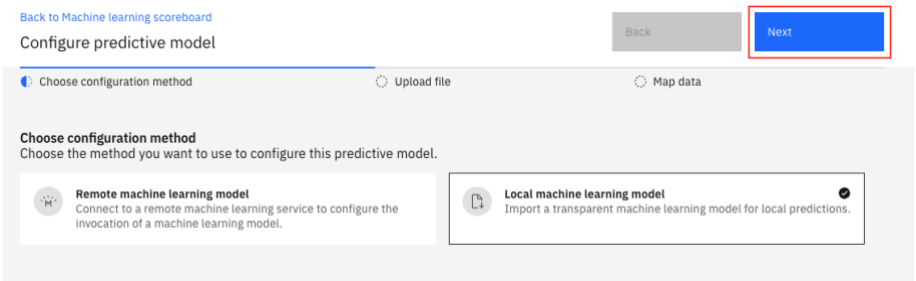
When you open the **Machine learning scoreboard** predictive model, you can see the model diagram and the different elements that have been pre-configured. Notice the **ML model invocation** node previously shown when a predictive model is first created has been replaced by four nodes marked with M. These nodes and the decision logic associated with them were automatically generated when the predictive model was connected to the local machine learning model defined in the **clientDefaultedPaymentRS.xml** file.



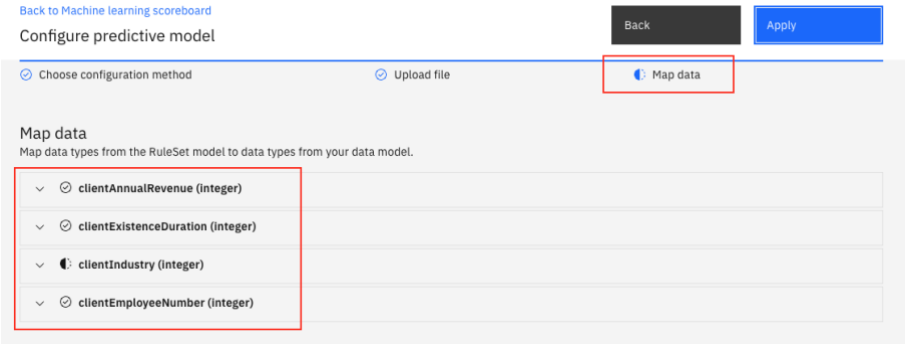
2. Click on the **Edit Configuration** button to explore the data mapping between the machine learning model and the data model used in the decision service.



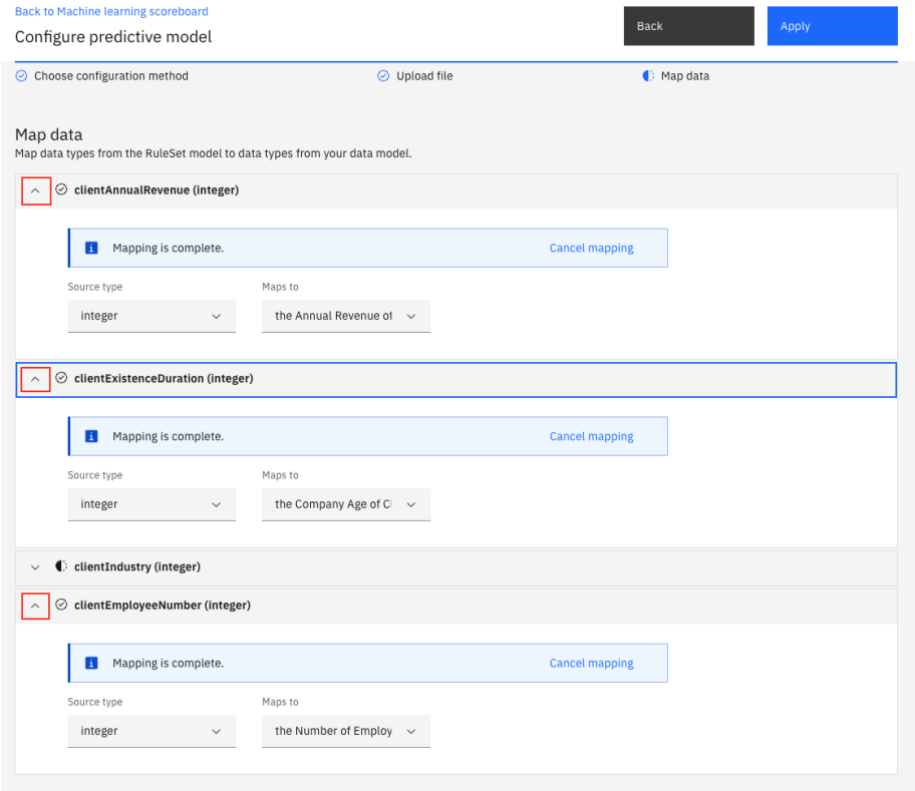
3. Once in the **Configure predictive model** pane, click the **Next** button until you see the **Map data** pane. Make sure you do **not** change the existing configuration while exploring.



4. On the **Map data** pane look at the four variable names displayed and their type. These variables, **clientAnualRevenue**, **clientExistenceDuration**, **clientIndustry**, **clientEmployeeNumber**, represent the input parameters for the machine learning model. They are defined in the **clientDefaultedPaymentRS.xml** file used to configure the local machine learning model.

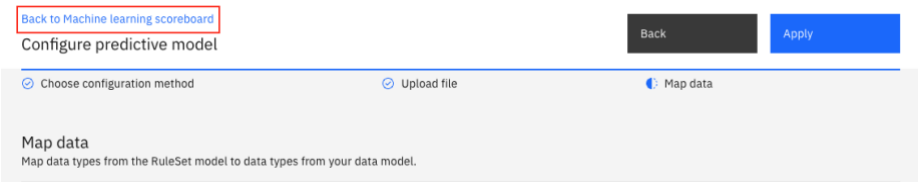


5. Expand the view for each of the variables by clicking on the downward arrow.

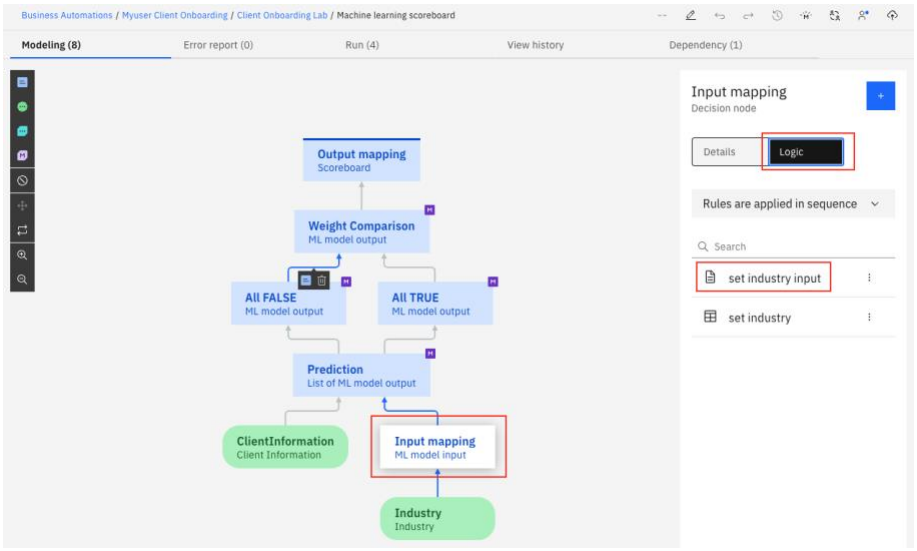


You see that three of the variables, **clientAnualRevenue**, **clientExistenceDuration** and **clientEmployeeNumber** have been mapped to a specific data type and variable from the decision service data model. The input variable **clientIndustry** is not mapped directly to the data model as a data type conversion needs to happen between the decision service data model and the data type expected by the machine learning data model. This conversion and the mapping happens in the **Input mapping** node.

6. Click on the **Back to Machine learning scoreboard** breadcrumb to go back to the predictive model.

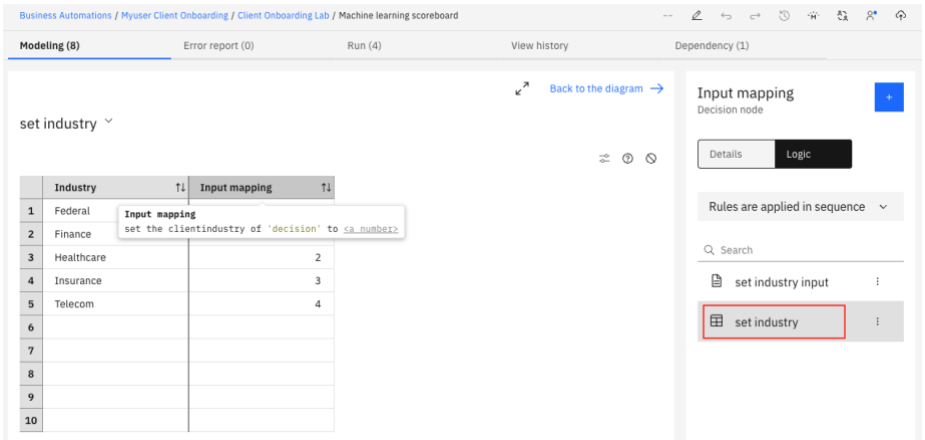


7. Click on the **Input mapping** node, then select the **Logic** tab on the right pane. Click on the **set industry input** business rule.

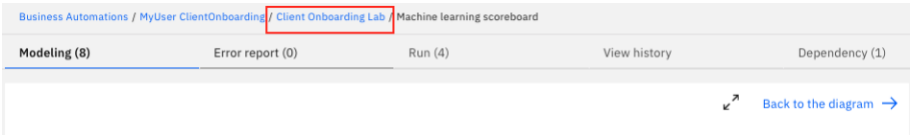


See how the input variable **clientIndustry** of the machine learning model is set to zero by the business rule. This value is set as the default but as you will see, it can be overwritten by the rules that follows.

8. Click now on the **set industry** decision table under the **Logic** tab. Since the **rule interaction policy** is set to **Rules are applied in sequence** the **clientIndustry** is set to a value from the **set industry** decision table if there is a match on the table conditions.



9. Click on the **Client Onboarding Lab** breadcrumb to navigate back to the **Models** tab for the decision service.

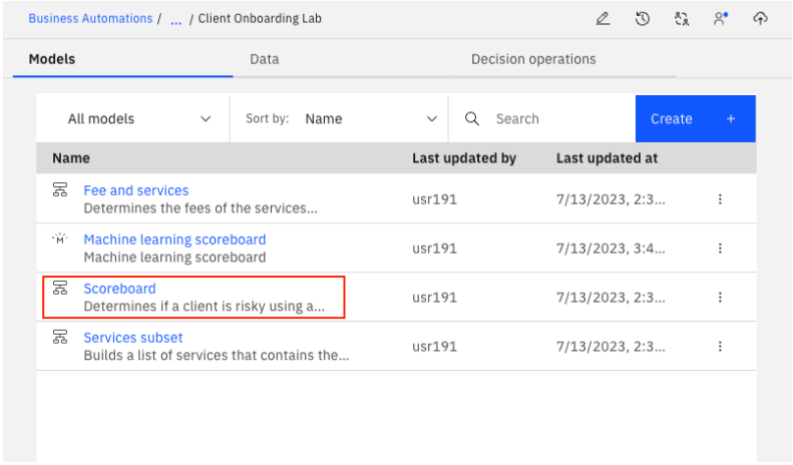


Now that we have explored the **Machine learning scoreboard** predictive model you will use it to infuse the results of the **Scoreboard** decision model. The **Scoreboard** decision model is used in the Client Onboarding decision service to calculate the client risk.

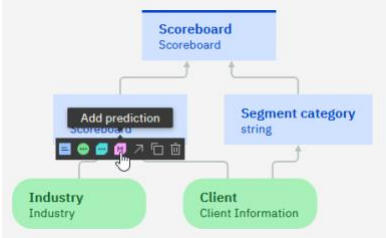
3.3 Use the Predictive Model

You will start by adding a prediction node to the **Scoreboard** decision model and connect it to the predictive model discussed in the previous section. You will also connect this prediction node to the **Predictive risk** input node and create a rule to define the client risk.

1. Click on the **Scoreboard** decision model.

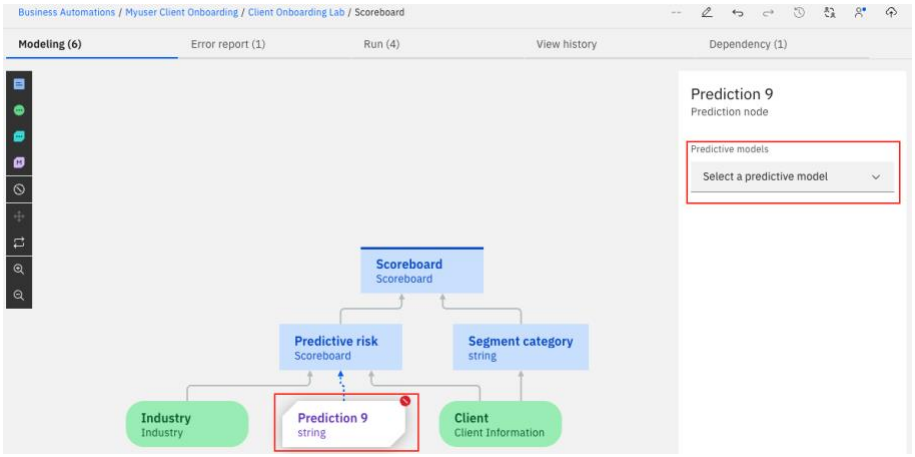


2. Once in the **Scoreboard** decision model, hover over the **Predictive risk** node and click on the **Add prediction** icon.

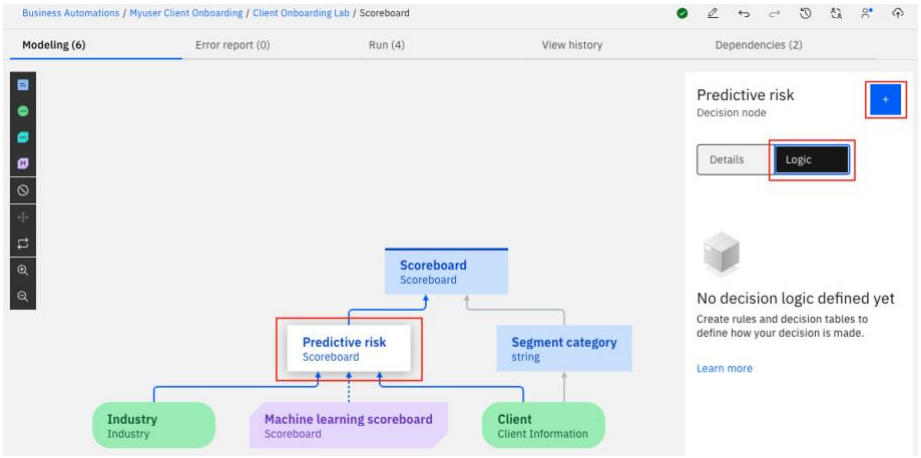


The prediction node will show an error because it needs to be connected to a predictive model.

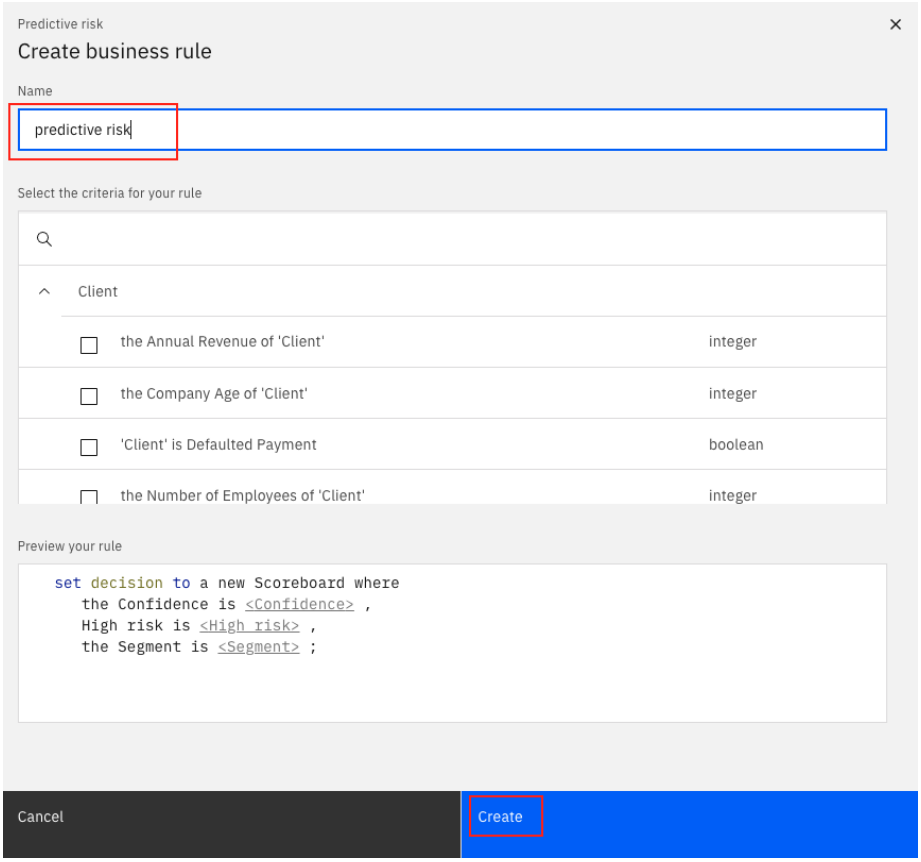
3. Click on the new **Prediction** node, and on the right-side pane select the predictive model, **Machine learning scoreboard**. This will connect the predictive model previously defined as input of the **Predictive risk** node.



4. You will now update the **Predictive risk** node with a rule that will invoke the **Machine learning scoreboard** predictive model to complete the client risk assessment. The Machine learning scoreboard will be called using the **Industry** and **Client** data available to the **Predictive risk** node. To complete this step, select the decision node **Predictive risk**. In the **Logic** tab, click the plus (**+**) sign and select the create **Business rule** option.



5. In the **Name** field enter **predictive risk** and click **Create**.

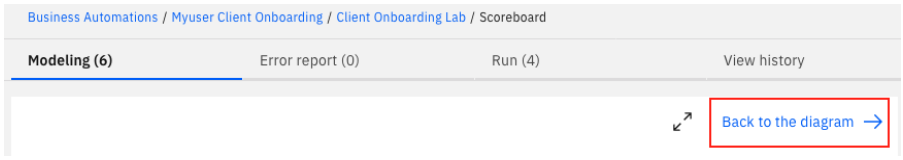


6. Once on the **predictive risk rule** editor, replace the full content of the rule with the text below. Use copy and paste to avoid errors.

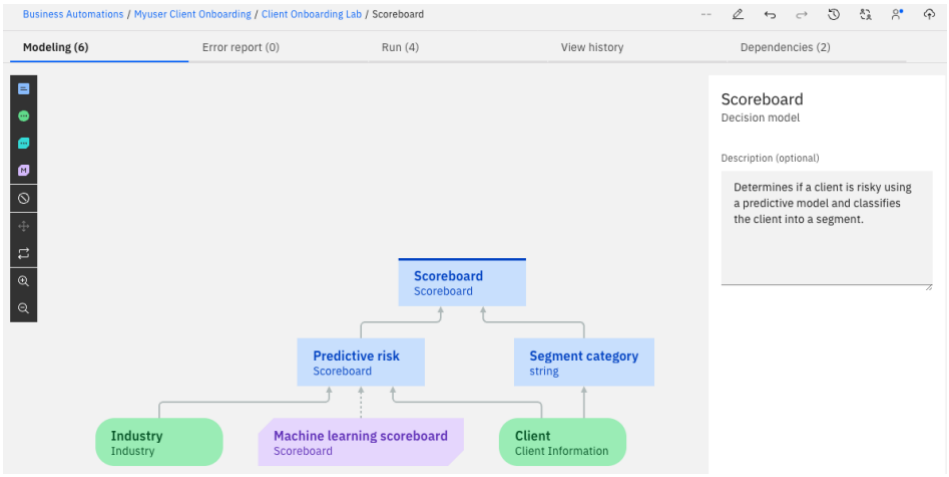
```
set decision to the machine learning scoreboard computed from
     ClientInformation being Client ,
     Industry being Industry ;
```

This rule sets the output of the **Predictive risk** node, of type Scoreboard, to the output of the **Machine learning scoreboard** using the **Client** and **Industry** information available to the node. In this step you are effectively infusing your **Scoreboard** decision model with the predictions from your machine learning model.

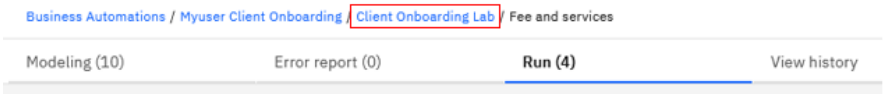
7. Click **Back to the diagram** to return to the **Scoreboard** decision model.



You have now completed the logic for the **Scoreboard** decision model leveraging the Machine learning scoreboard provided in the Lab.



8. Click on the **Client Onboarding Lab** breadcrumb to navigate back to the **Models** tab.



3.4 Summary

You have completed the Adding Machine Learning to a Decision Model exercise. You explored the elements of a predictive model backed by an underlying machine learning model and used it to calculate the client risk for the Client Onboarding scenario.

4 Exercise: Sharing and Deploying Decision Services

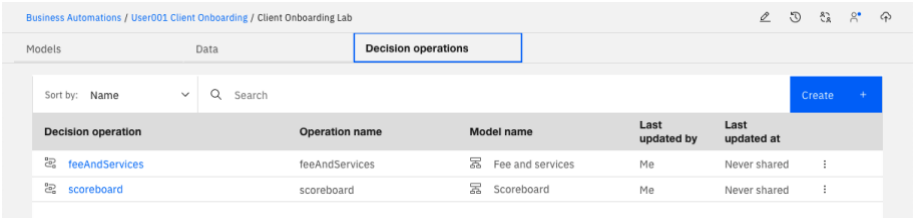
4.1 Introduction

In this exercise, you learn how to collaborate in the development of your decision service and how to make it available for execution by other components within or outside the IBM Cloud Pak for Business Automation platform. You will connect your decision project to a Git repository and will deploy the decision service as a decision archive in the Automation Decision Services embedded runtime.

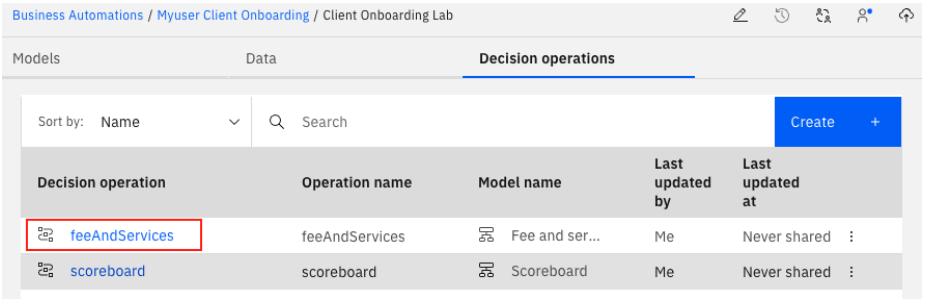
4.2 Exploring a Decision Operation

Decision operations are entry points to a decision, task or predictive model. They allow you to externally access the decision logic in your models as APIs when a decision service is deployed. A decision operation includes a reference to a decision model and the input and outputs that must be used to call the model. A decision service must contain at least one decision operation to be deployed. The decision service prepared for the Lab contains two decision operations **feeAndServices** and **scoreboard**. Both operations map to the decision models you previously worked on, **Fee and services** and **Scoreboard** respectively. When you deploy the decision service in this lab each of these operations will be available for you to call externally.

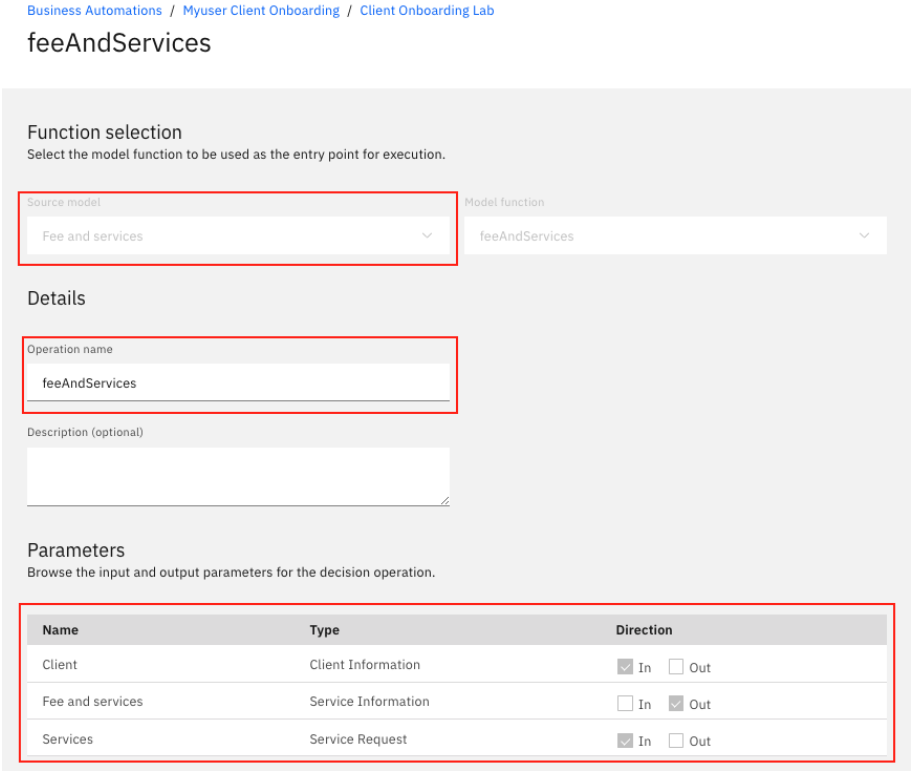
1. Open the **Decision operations** tab to explore the Decision operations.



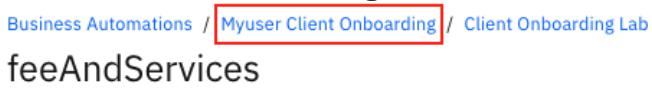
2. Click on the **feeAndServices** decision operation.



3. Inspect the **feeAndServices** decision operation. Look at the **Function selection**, **Details** and **Parameters** sections. The decision logic for this decision operation comes from the **Fee and services** decision model, the name assigned to the decision operation is **feeAndServices** (the name of the operation does **not** have to match the name of the model used) and the input and output parameters of the service map to the input and output parameters of the decision model used.



4. Click on **<user> Client Onboarding** in the breadcrumbs to return to the project.



Now that you explored the decision operations available in the Client Onboarding decision service, you will share your decision service. You will start by creating a git repo.

4.3 Creating a Git repository on GitHub

While other Git implementations are supported, in this lab you will use GitHub to host the repository required to share your decisions project. **You must have a GitHub account** to complete this step and the rest of this exercise. If you do not have a GitHub account, create one at github.com.

To create a repository for this lab:

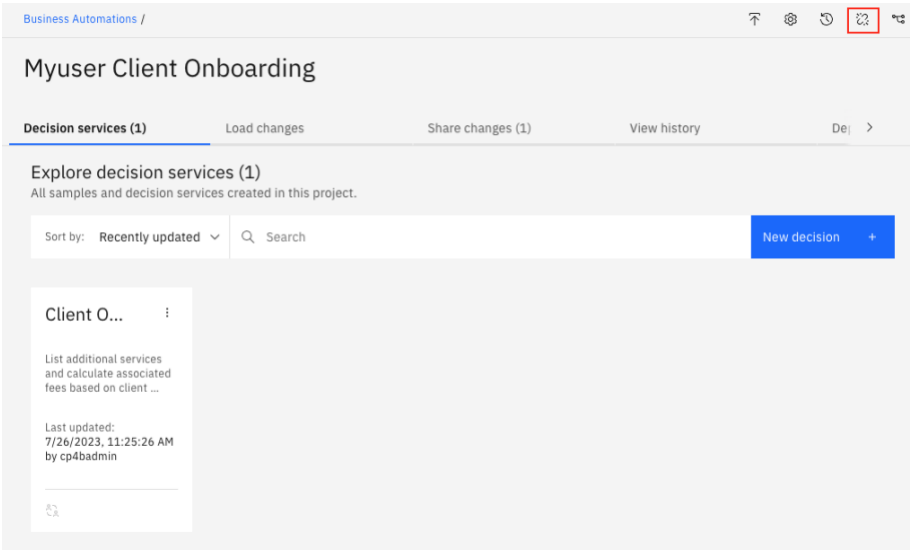
1. Open GitHub in your browser, and sign-in with your GitHub credentials.
2. Click the plus (**+**) button at the upper-right part of the page and select **New repository** to create an empty repository.
3. Give the repository a unique name, and add the following description:
   Git repository for the Automation Decision Services Lab
4. Select Private and Click on **Create repository**. The repository must **not** contain a **readme**, **.gitignore**, or **license** file.
5. In the **Quick setup** section click on the **HTTPS** button.

6. To the right side of the **HTTPS** URI field, click the copy button, then paste the value in a safe location for use in the next set of steps. The URI has the following format: https://github.com/<yourAccountName>/<yourRepoName>.git
7. Open the drop-down list for your GitHub account in the upper right corner of the page.
8. Click **Settings** and then **Developer settings** > **Personal access tokens**.
9. Click on **Generate New Token** in the upper-right corner. You can choose either **Fine-grained tokens** or **Tokens (classic)** based on your preference although **Tokens (classic)** is easier.
10. Enter a name, and make sure it has access to the **repo** scope permissions to give full control of the repository you just created to Automation Decision Services.
11. Click **Generate token** at the bottom of the page. Copy the generated access token before closing this page, then paste the value in a safe location for use in the next set of steps.

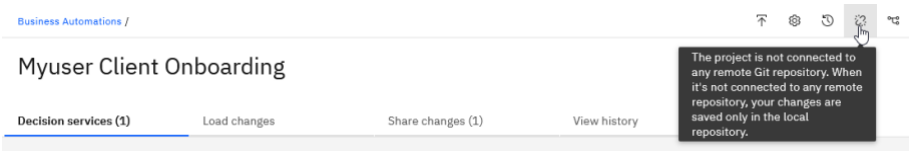
4.4 Connecting Your Project to the Git Repository

You will now connect your project to the git repository previously created. The git repository will allow you to keep the history of all changes made to your decision service from Automation Decision Services. Connecting to a Git repository is required to be able to build and deploy decision services.

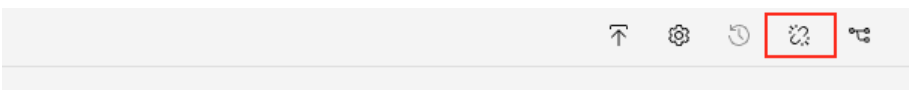
1. Check the status of the **Remote Git repository** in the upper right corner of Decision Designer by hovering over the Git connection button.



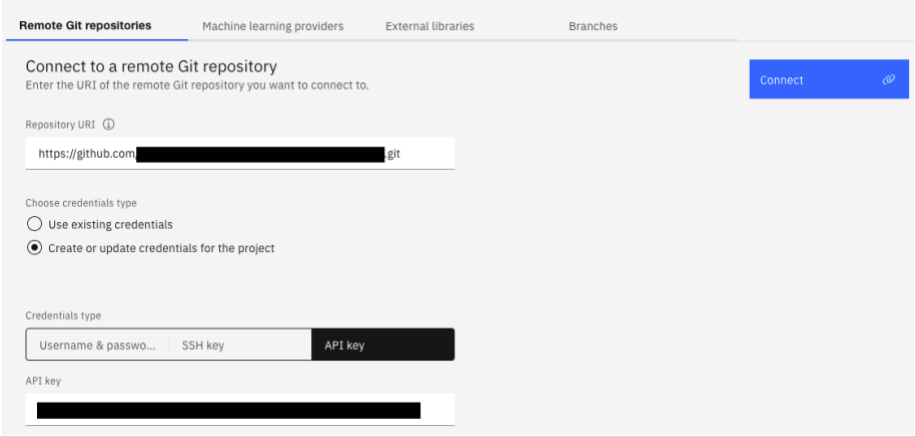
It shows that the project is not connected.



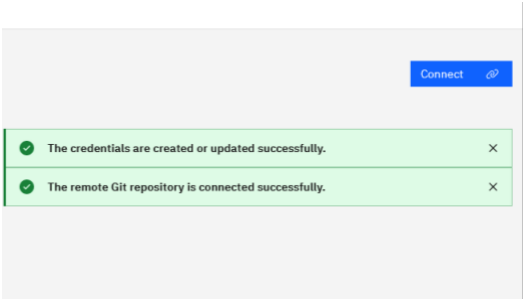
2. Click on the **Remote Git repository** connection button to configure.



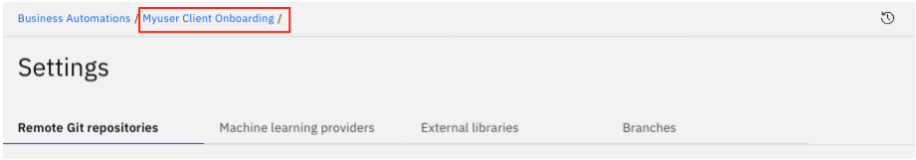
3. Enter the Git repo URI you previously saved, select **API key**, and enter the personal access token you previously saved, then click the **Connect** button.



Upon successful connection, Decision Designer displays the following messages and updates the status of the Remote Git repository.



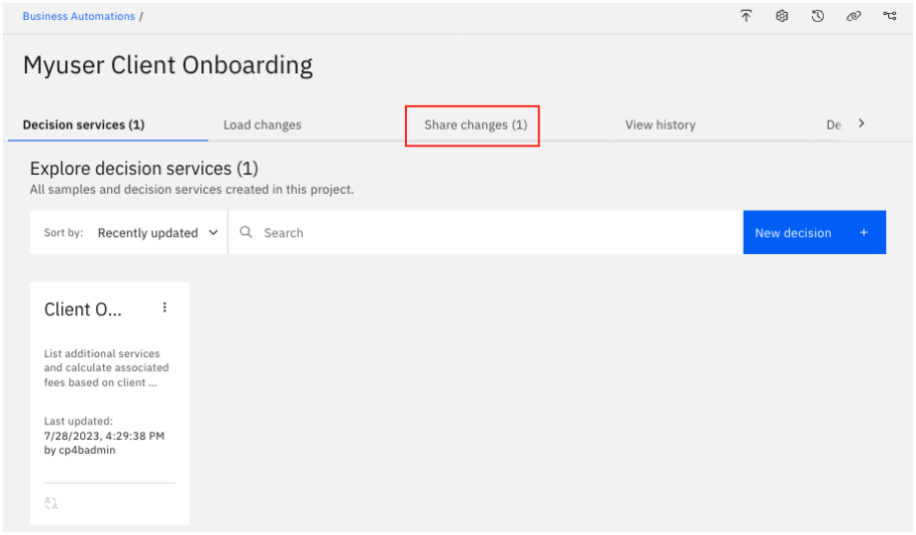
4. Click on **<user> Client Onboarding** in the breadcrumbs to return to the project.



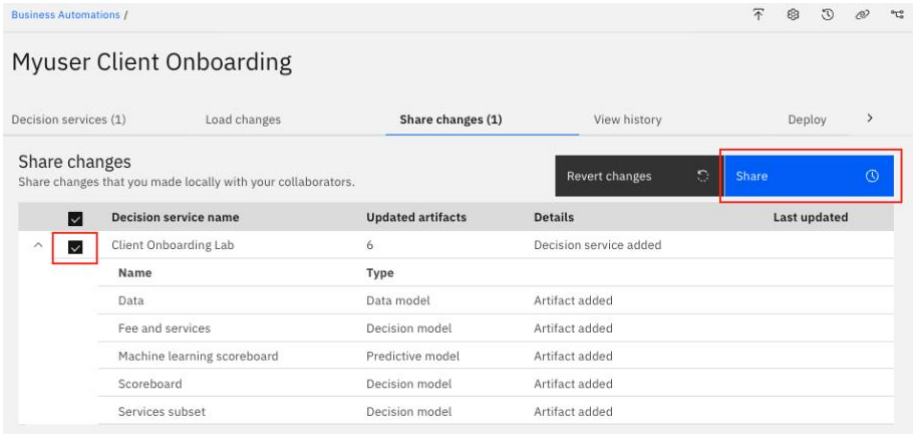
4.5 Sharing Your Decision Service

The changes made on a decision service become visible when you share them. You give the access rights in Business Automation Studio to your collaborators. To share the **Client Onboarding** decision service:

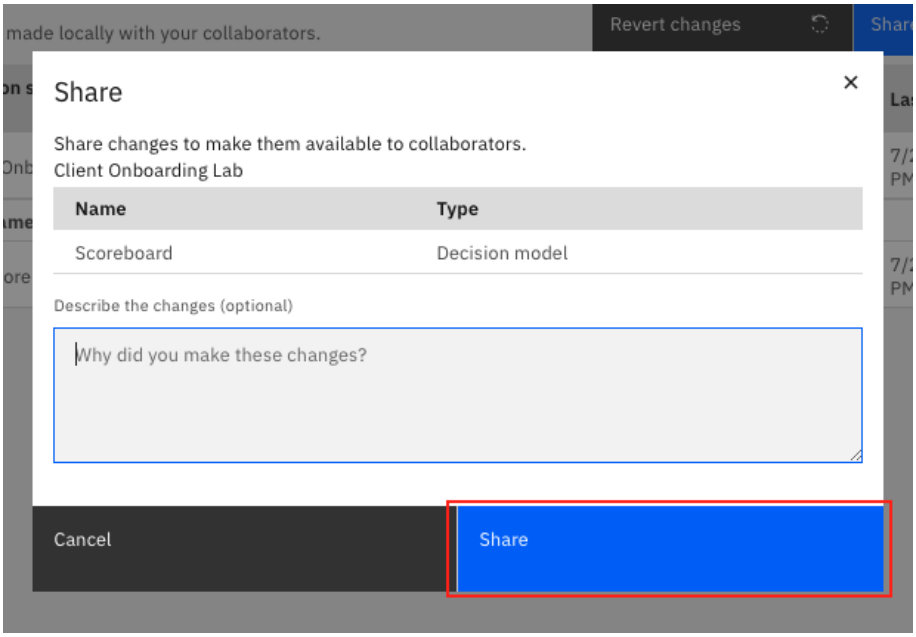
1. Click on the **Share changes** tab.



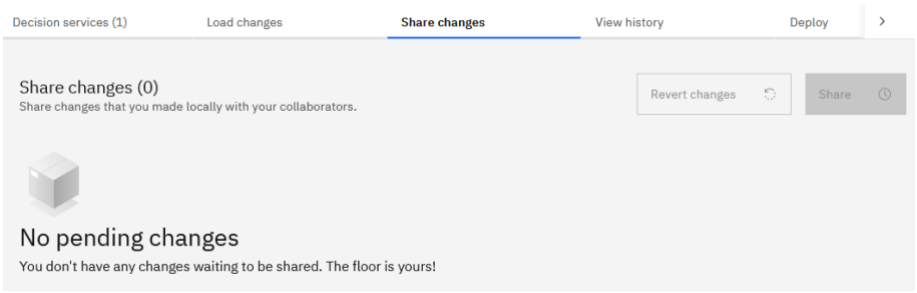
2. Ensure the checkbox next to **Client Onboarding Lab** is selected and click **Share**. The number of changes in your project may be slightly different.



3. Enter a comment to describe the changes and then click **Share**.



4. After the changes are shared, you see that there are no pending changes.

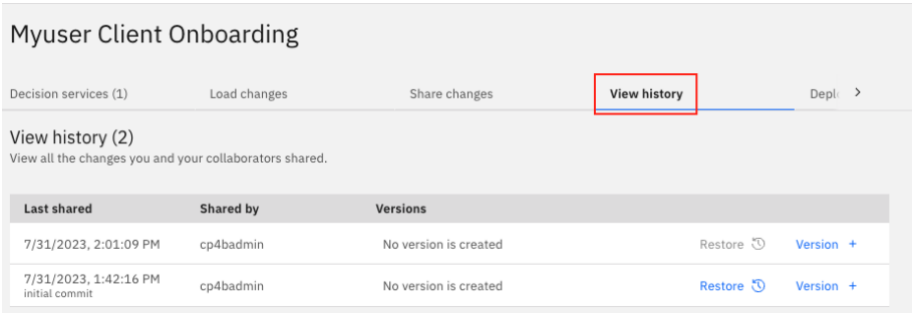


Sharing changes means that updates done locally are published in your Git repository and can be visible to other users via Business Automation Studio once you give them access. Providing access to other users via Business Automation Studio is outside of the scope of this lab. For additional information and procedures see Managing access to decision automations.

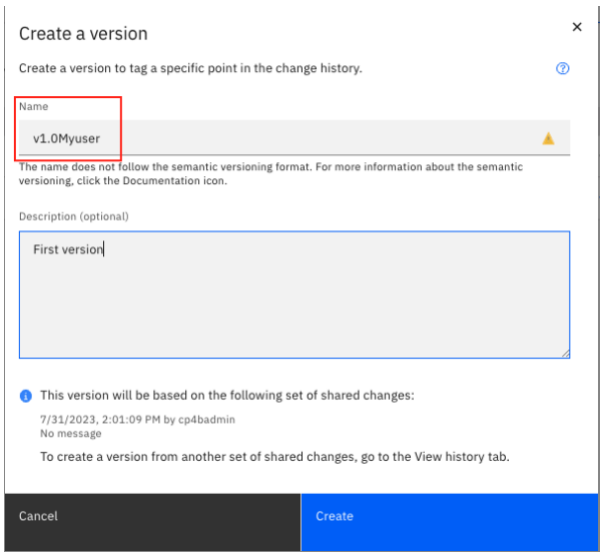
4.6 Creating a Version

To deploy your decision service, you need to create a version of the current content. Versions correspond to tags in Git. A version is a snapshot of the project and records a point-in-time of the decision services within the project.

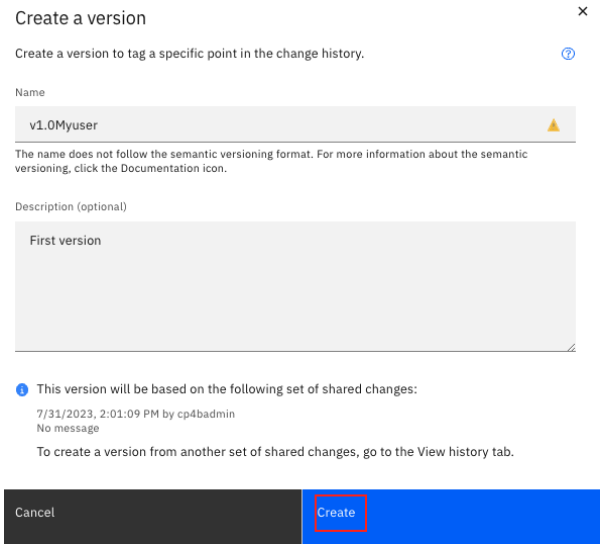
1. Click on the **View history** tab. The View history pane shows all the changes made to the project. You may have a slightly different amount of history for your project.



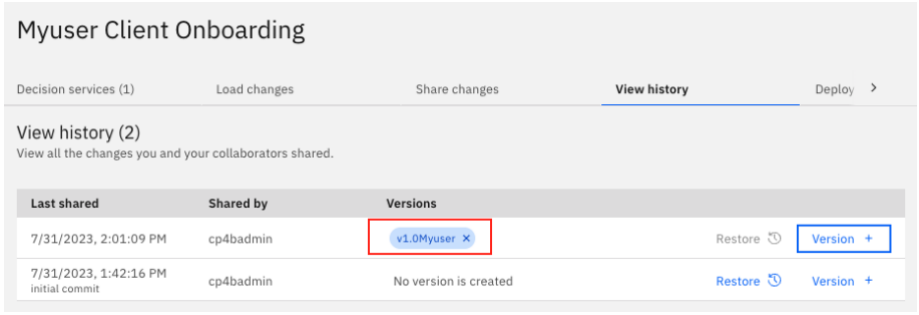
2. In the row representing the most recent version, click the **Version +** button to the right.
3. Enter a name for this version. For lab purposes it is recommended to add your username as part of the version name such as **v1.0<**YourUser**>**.



4. Click **Create**.



A new version is created.

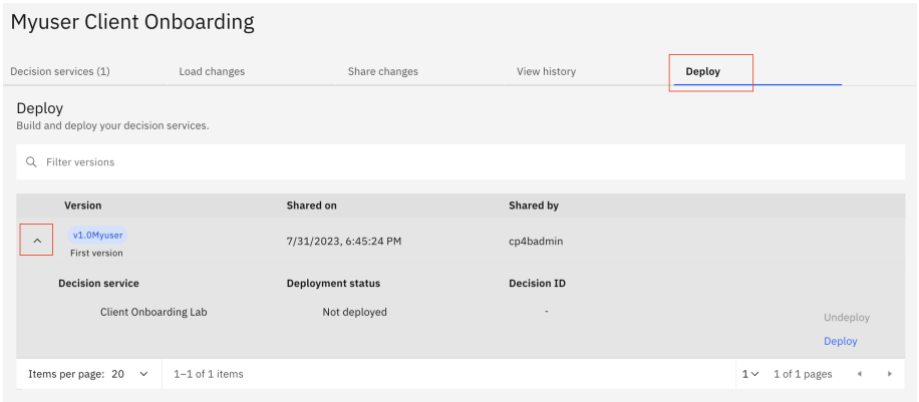


You are now ready to deploy your version.

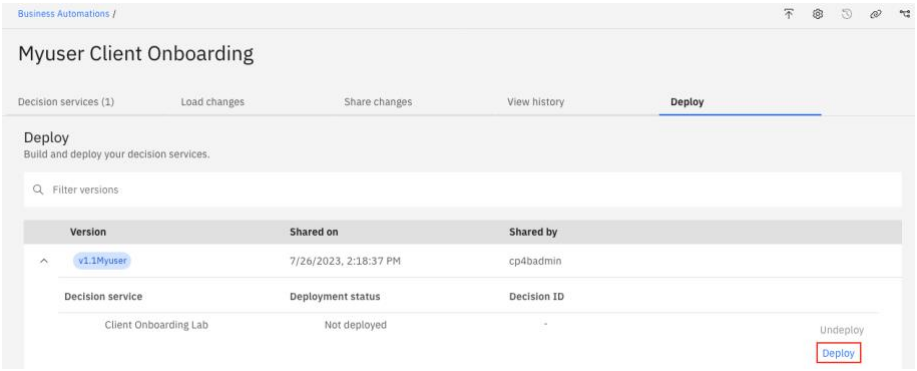
4.7 Deploying Your Project

Now that you have created a version, you can build and deploy your decision service directly in Automation Decision Services. A decision service archive is built from your decision service and is deployed to an embedded decision runtime.

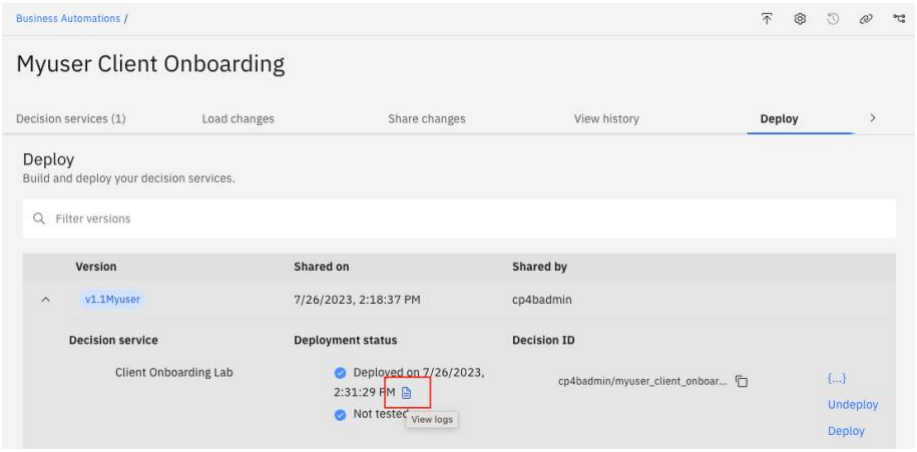
1. Click the **Deploy** tab and expand the version your created in the previous section.



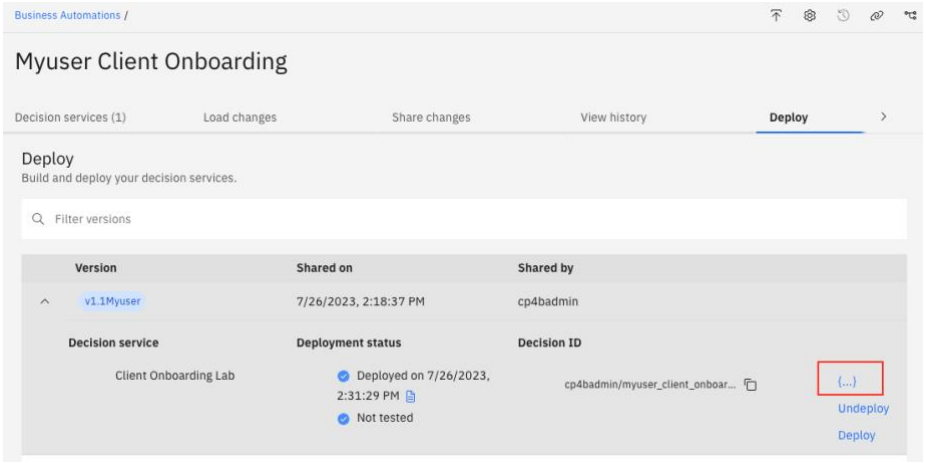
2. Click the **Deploy** link in the Deploy pane and then click the **Deploy** button in the confirmation window. This triggers a build and a deployment of the decision service into the embedded runtime available in Automation Decision Services. Wait for the deployment to finish, this could take a few minutes.



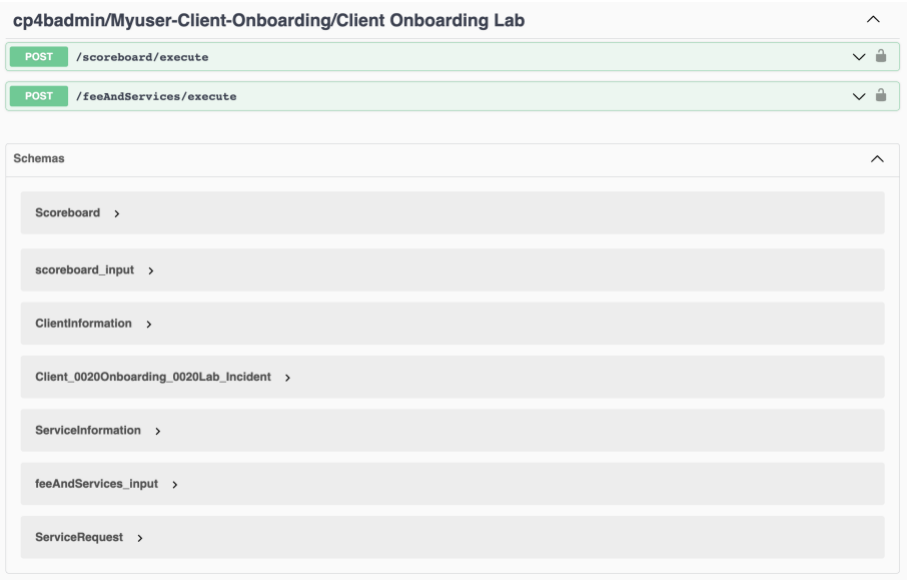
3. Once completed, you will see the deployment details under the **Deployment status** section. You can check the logs by clicking the **View logs button**.



4. To explore the APIs available in your decision service, click the **{...}** link next to the **Decision ID**.



This will open a Swagger UI where you can explore and execute the decision operations available as REST APIs. Notice that these REST APIs map to the decision operations that we discussed in the Exploring a Decision Operation section. The Swagger interface also lets you explore the data model used for the REST API.



4.8 Summary

In this exercise you made the Client Onboarding decision service visible for other collaborators by sharing the changes. You also connected the decision project to a Git repository and explored the use of decision operations to define entry points for the **Fee and services** and **Scoreboard** decision models. Finally, you were able to deploy the **Client Onboarding** decision service into the Automation Decision Services embedded runtime and inspected the REST APIs available in the decision service.

Congratulations on completing the lab!