



Lab Guide: Introduction to AI Governance in watsonx

Author: Elena Lowery, elowery@us.ibm.com



Contents

Overview.....	3
Required software, access, and files	3
AI Governance and watsonx.governance	4
Watsonx.governance	5
Part 1: Create a Model Inventory	6
Part 2: Prompt Development and Evaluation.....	11
Conclusion	38



Overview

In this lab you will learn how to track and evaluate large language model (LLM) prompts in *watsonx.governance*. We will complete the following tasks:

- Create a model inventory
- Create a use case
- Test and evaluate a prompt
- Promote the prompt to production
- Use the prompt from a client application.

Required software, access, and files

Note: if you have completed these steps in the previous labs, you don't need to repeat them.

- To complete this lab, you will need access to *watsonx.ai*. You can get access by signing up for an [IBM Cloud account](#) and provisioning *watsonx.ai* and *watsonx.governance* service.
- A Python IDE with Python 3.10 environment (*Visual Studio Code* or *PyCharm*)
- You will also need to download sample prompts and code samples from [this folder](#).
- Unzip the downloaded folder if it was zipped up during download. In the lab we will refer to this folder as *lab repo*.

Important note: Some screenshots in the lab may be slightly different from the product. If you have questions, please ask your workshop instructor.



AI Governance and watsonx.governance

AI Governance is a discipline that includes business process and implementation of best practices for *creating, deploying, managing, and monitoring* AI artifacts, such as Machine Learning (ML) models and Large Language Models (LLMs).

In the past, components of *AI Governance* were implemented by companies in highly regulated industries or for specific use cases. For example, companies in the financial industry must have Model Risk Governance (MRG) practice to comply with regulatory requirements that were first [published](#) in 2012, and even as early as 2000. However, in some cases the implementation of *AI Governance* was simply keeping track of models in a spreadsheet or a document.

During the past few years various government organization proposed and approved laws for governing AI. Here's an example of a law related to the use of AI:

New York



Enacted

In December 2021, New York City passed the first law (Local Law 144), in the United States [requiring employers to conduct bias audits of AI-enabled tools used for employment decisions](#). The law imposes notice and reporting obligations.

Specifically, employers who utilize automated employment decision tools (AEDTs) must:

1. [Subject AEDTs to a bias audit](#), conducted by an independent auditor, within one year of their use;
2. Ensure that the date of the most recent bias audit and a "summary of the results", along with the distribution date of the AEDT, are publicly available on the career or jobs section of the employer's or employee agency's website;
3. Provide each resident of NYC who has applied for a position (internal or external) with a notice that discloses that their application will be subject to an automated tool, identifies the specific job qualifications and characteristics that the tool will use in making its assessment, and informs candidates of their right to request an alternative selection process or accommodation (the notice shall be issued on an individual basis at least 10 business days before the use of a tool); and
4. Allow candidates or employees to request alternative evaluation processes as an accommodation.

[Source](#)

Regulatory landscape for AI continues to evolve, with important regulations such as the United States [AI Bill of Rights](#) and [EU AI Act](#) will likely accelerate development of guidelines and laws.

One of the best ways to prepare for upcoming regulations is to infuse *AI Governance* in several steps of the ML model and LLM lifecycle. *Watsonx* includes multiple features that

will streamline not only governance of models, but also improve operationalization and simplify maintenance.

In this lab we will focus on *operational* aspects of implementing AI Governance. We will cover *Model Risk Governance (MRG)* and automatic integration of MRG with model lifecycle in a separate lab.

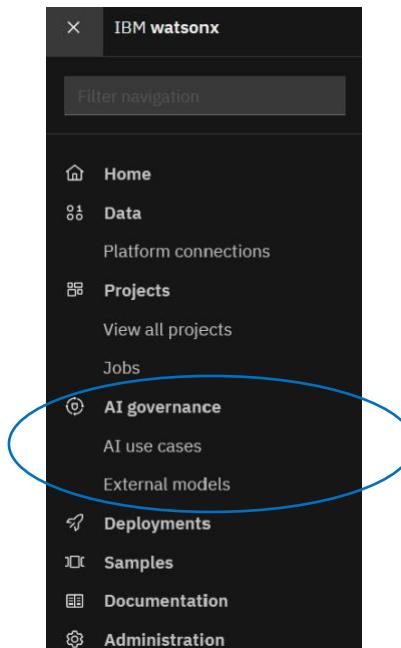
Watsonx.governance

Many software offerings today are packaged as individual services that include a set of features. Services provide granular capabilities that can be included in multiple offerings. IBM's *watsonx.governance* includes two services:

- **AI Factsheets:** used to capture model and prompt metadata as well as track model lifecycle
- **OpenScale:** used to monitor model performance, accuracy, provide explainability, and alert for bias.

It's common in microservice software design to have prerequisite services. In case of *watsonx.governance*, the prerequisite services are *watsonx.ai* and *Watson Machine Learning*.

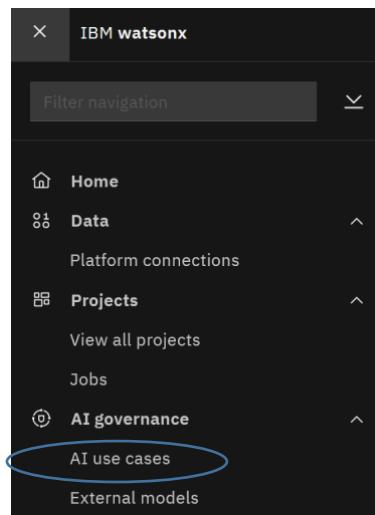
From the end-user perspective, all services are seamlessly integrated into *watsonx* UI. As additional services are provisioned, new actions (menus/buttons) become available in various components of *watsonx*. In other words, end users don't need to know the names of the services, but this information may be useful to administrators. For example, if a user doesn't see a certain action, the administrator can check if the required service was provisioned.



Part 1: Create a Model Inventory

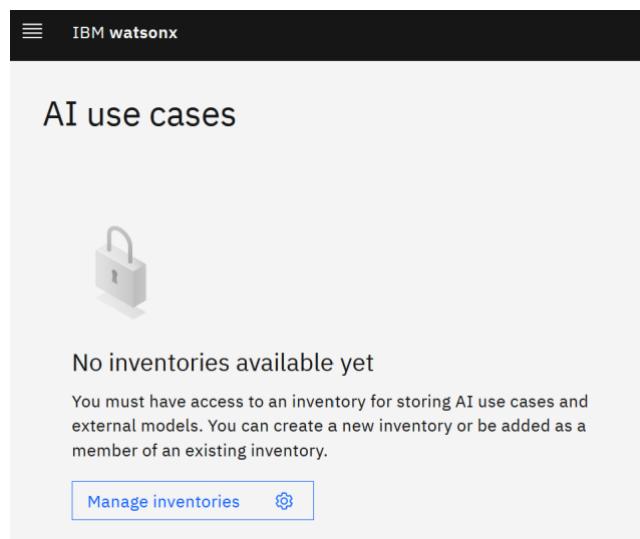
Creating a *Model Inventory* is the first step in better organization of AI artifacts.

1. Log in to *watsonx* in your browser.
2. From the main menu in the top left corner select **AI Use Cases**



Note: If you don't see this menu, check that OpenScale service was provisioned in your IBM Cloud account.

3. If you have not previously created a Model Inventory, you will see an option to create it. Click **Manage Inventories**.

A screenshot of the AI use cases page. The title 'AI use cases' is at the top. Below it is a large lock icon. The text 'No inventories available yet' is displayed. A message below states: 'You must have access to an inventory for storing AI use cases and external models. You can create a new inventory or be added as a member of an existing inventory.' At the bottom is a blue button labeled 'Manage inventories' with a gear icon.



4. Click on the **New Inventory** button, then provide a unique name for your inventory, for example, *LLM Insurance Use Cases <your initials>*. Make sure to select *Cloud Object Storage* associated with your account id.

Click **Create**.

Optionally, add collaborators and close the screen.

The screenshot shows the 'New Inventory' dialog box. At the top right is a blue button labeled 'New inventory +'. The main area has two sections: 'Name' (containing 'LLM Insurance Use Cases EL') and 'Description (optional)' (containing 'Inventory for tracking LLM use cases for the insurance industry'). To the right, there's a sidebar for 'IBM Cloud Object Storage' which says 'Object storage instance' and 'Cloud Object Storage-ur', with a link 'Or create a new Cloud Object Storage instance'. At the bottom left is a checked checkbox for 'Add collaborators after creation' with the note 'An inventory without collaborators will be visible to only you. Add members to collaborate on AI use cases.'

Your inventory will look similar to the following screenshot:

Inventories			
Add or modify an inventory to manage access and storage for a collection of AI use cases and external models.			
Name	Date created	Creator	Your role
LLM Insurance Use Cases EL	1 minute ago	EL Elena Lowery	Admin

Next, we will create an AI use case.

5. Navigate to the **AI Use case** menu again and click the **New AI use case** button.

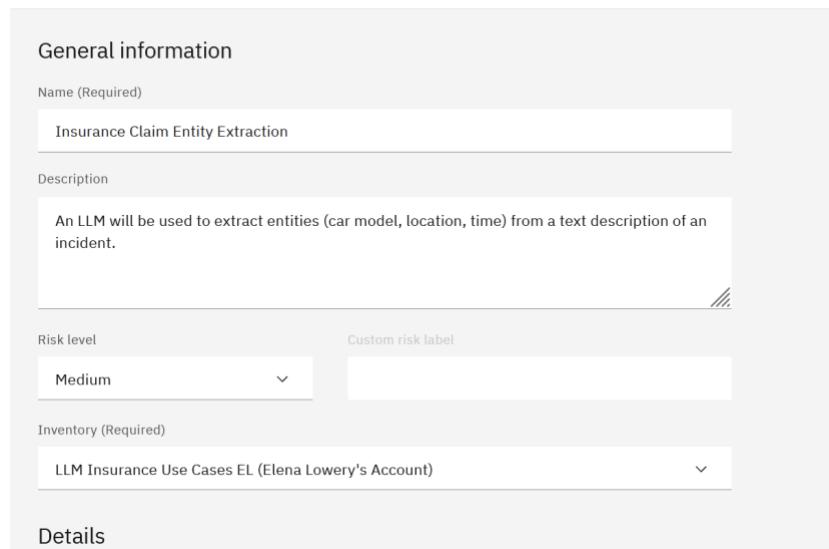
The screenshot shows the 'AI use cases' list page. At the top right is a blue button labeled 'New AI use case +'. The main area has a search bar 'Find a AI use case' and a table with columns: Name, Status, Owner, Inventory, Tags, Risk level, and Alerts in.

6. Review metadata options available on this screen.

First, we provide use case information, such as *name* and *description*, as well as *risk level* and *inventory name*. Risk level designation can be based on industry regulations or an organization's evaluation of risk. It can also help with prioritization of work if multiple models perform below specified thresholds.

New AI use case

Create a use case to define a business problem, request a model, and specify details such as risk level and status.



The screenshot shows the 'General information' section of the 'New AI use case' form. It includes fields for Name (Required), Description, Risk level (Medium), Custom risk label, Inventory (Required), and Details.

General information

Name (Required)
Insurance Claim Entity Extraction

Description
An LLM will be used to extract entities (car model, location, time) from a text description of an incident.

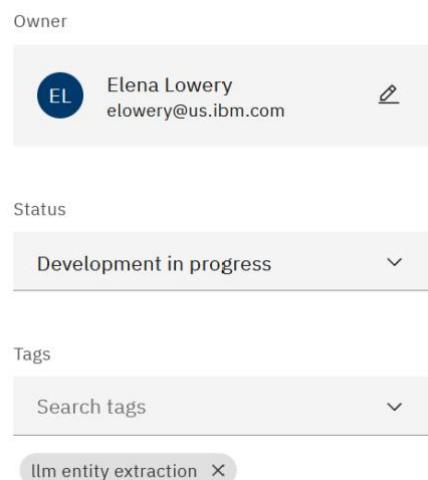
Risk level
Medium

Custom risk label

Inventory (Required)
LLM Insurance Use Cases EL (Elena Lowery's Account)

Details

On the right side, we can capture some aspects of model lifecycle process, specifically the status, which we decided to set to *Development in Progress*. Adding tags to the use case entry will help us easily find use cases in the inventory.



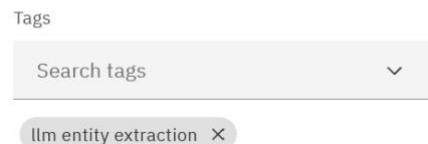
The screenshot shows the 'Owner' and 'Status' sections of the use case form. It includes a profile picture for Elena Lowery, her email (elowery@us.ibm.com), a edit icon, and a dropdown menu for Status (Development in progress).

Owner

Elena Lowery
elowery@us.ibm.com

Status

Development in progress



The screenshot shows the 'Tags' section of the use case form. It includes a search bar for 'Search tags' and a tag labeled 'llm entity extraction' with a delete icon.

Tags

Search tags

llm entity extraction X



Create a use case record using the sample values in the screenshot or provide your own.

The detailed *Use Case* view provides the summary of details we entered.

The screenshot shows the 'AI use cases' interface for 'Insurance Claim Entity Extraction'. The 'Overview' tab is selected. On the left, there's a sidebar with 'Discover next steps' and a 'General information' section. The main area displays the following details:

Name	Status
Insurance Claim Entity Extraction	Development in progress (Elena Lowery, Dec 28, 2023)
Description	Risk level
An LLM will be used to extract entities (car model, location, time) from a text description of an incident.	Medium
Owner	Inventory
Elena Lowery elowery@us.ibm.com	LLM Insurance Use Cases EL
Tags	
llm entity extraction	

If you click on the *Lifecycle* tab, you will notice that we don't have any information yet because we have not added project assets that we want to track.

The screenshot shows the 'AI use cases' interface for 'Insurance Claim Entity Extraction'. The 'Lifecycle' tab is selected. The main area displays the following sections:

- Welcome to your new AI use case**: Shows icons for team work and assets.
- Discover next steps**: Shows a section for organizing team work with approaches.
- Organize team work with approaches**: Describes how to use approaches to organize assets.
- Manage AI assets with versions**: Describes how each approach starts a new version series for identifying associated assets.

At the bottom, there's a section for 'Default approach' which states: 'A default approach for tracking your AI assets.' and 'No AI assets tracked in this approach.'



7. Create a second use case entry for the *Summarization* use case. You can use the values in the following screenshot or provide your own.

New AI use case

Create a use case to define a business problem, request a model, and specify details such as risk level and status.

General information

Name (Required)

Description

Summarization of an submitted insurance claim using an LLM

Risk level

Custom risk label

Low

Inventory (Required)

LLM Insurance Use Cases EL (Elena Lowery's Account)

When finished, your AI use case view should look similar to this screenshot.

AI use cases							
Name	Status	Owner	Inventory	Tags	Risk level	Alerts in	
Insurance Claim Summarization	Development in progress	Elena Lowery	LLM Insurance Use Cases EL	llm summarization	Low	none	
Insurance Claim Entity Extraction	Development in progress	Elena Lowery	LLM Insurance Use Cases EL	llm entity extraction	Medium	none	

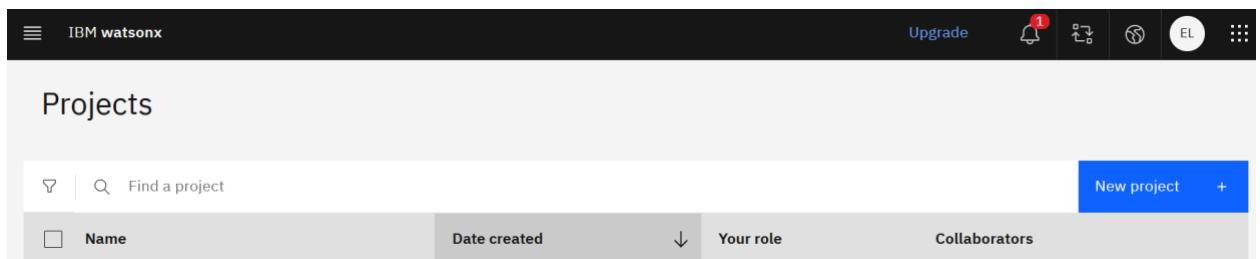
We have completed setting up a *Model Inventory* and creating metadata for AI use cases that we will implement with LLMs. In just a few minutes, we provided a better organization for auditing, tracking, and monitoring AI assets that will be used in various applications.

Part 2: Prompt Development and Evaluation

In this section we will use provided prompts for the defined use cases to walk through the lifecycle of deploying a prompt/LLM for use by production applications. We will not cover the iterative process of prompt engineering and prompt tuning. Since we're focusing on governance, we are making the assumption that the development process for the prompts has been completed.

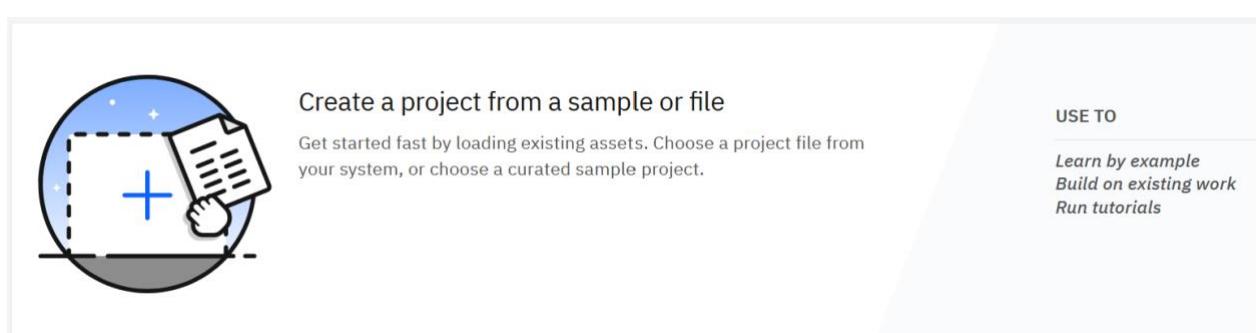
First, we will import the project that contains prompts into watsonx.

1. In *watsonx* navigate to the **Projects** view (from the main menu in the top left corner) and click the **New Project** button.



The screenshot shows the 'Projects' view in IBM WatsonX. At the top, there's a navigation bar with the IBM WatsonX logo, an 'Upgrade' button, and several icons. Below the navigation is a search bar with the placeholder 'Find a project'. To the right of the search bar is a blue button labeled 'New project +' with a '+' icon. Underneath the search bar is a table with four columns: 'Name', 'Date created', 'Your role', and 'Collaborators'. The 'Name' column has a blue header bar. The rest of the table has grey header bars.

2. Click the **From file** option and navigate to the downloaded lab repo */Cloud Projects* folder to select *Insurance_LLM_Use_Cases.zip*.



The screenshot shows a dialog box titled 'Create a project from a sample or file'. It features a large circular icon on the left containing a hand holding a document with a blue plus sign on it. To the right of the icon, the text reads 'Create a project from a sample or file' and 'Get started fast by loading existing assets. Choose a project file from your system, or choose a curated sample project.' On the far right, there's a section titled 'USE TO' with three items: 'Learn by example', 'Build on existing work', and 'Run tutorials'.

Provide a unique project name and click **Create**.

IBM watsonx

Upgrade 1 | Cloud Object Storage | Watson Studio | Watson Assistant

Create a project from sample or file

[From a file](#) [From sample](#)

Select file ✓

Choose a .ZIP file that contains an exported Watson Studio project.

Insurance_LLM_Use_Cases.zip X

Name
Insurance LLM Use Cases EL

Description (optional)
A collection of prompts, data, and notebooks for insurance use cases implemented with LLMs

Storage

Project includes integration with [Cloud Object Storage](#) for storing project assets.

Cloud Object Storage-ur

Controls

Cancel Back Create

3. Switch to the **Manage** tab, then select **Services and Integrations** tab. Click **Associate Service**.

IBM watsonx

Projects / Insurance LLM Use Cases EL

Upgrade 1 | Cloud Object Storage | Watson Studio | Watson Assistant

Overview Assets Jobs Manage

Project

- ⚙️ General
- 🔗 Access control
- 🌐 Environments
- 📊 Resource usage
- ⚙️ Services & integrations +

Services & integrations

IBM services Third-party integrations

Find services Associate service +

Name	Service type

4. Select the displayed **Machine Learning** service and click **Associate**.

Projects / Insurance LLM Use Cases EL

Overview Assets Jobs Manage

Project

- General
- Access control
- Environments
- Resource usage
- Services & integrations**

Services & integrations

IBM services (1) Third-party integrations

Find services Associate service +

Name	Service type
Machine Learning-fz	Watson Machine Learning

5. Switch to the **Assets** tab and expand **Prompts**.

Projects / Insurance LLM Use Cases EL

Overview **Assets** Jobs Manage

Find assets

7 assets All assets

Asset types

- Data (4)
- Prompts** (3)

Prompts

Name	Last modified
Insurance claim suggested next steps Prompt template	2 minutes ago Modified by Service
Insurance claim key information extraction Prompt template	2 minutes ago Modified by Service
Insurance claim summarization Prompt template	2 minutes ago Modified by Service

Let's review prompts for our AI use cases.

6. Click on the *Insurance claim key information extraction* prompt. We are making an assumption that you're familiar with prompt structure, but let's point out a few things that may be unique to prompt format in the *Prompt Lab*.

Note: when prompted, check all the boxes to agree to terms and click *skip tour*.



Welcome to Prompt Lab

By using any foundation model provided with this Cloud Service, you acknowledge and understand that:

- Some models included in the Cloud Service are Non-IBM Products. Review the applicable model details on the third party provider and license terms that apply.
- Models may generate outputs that contain misinformation, obscene or offensive language, or discriminatory content. Client should review the outputs for such information or content prior to reuse. Users should review and validate the outputs generated.
- The output generated by all models is provided to augment, not replace, human decision-making by the Client.

[Skip tour](#) [Start tour](#)

When you open the prompt, you can choose to open in preview mode or edit mode. Edit mode will lock the prompt and allow you to make changes. Choose *Edit* for now.

Edit this prompt template?

If you open this prompt template in edit mode, you lock it for other users. You can preview the prompt template without locking it.

Don't show this message again.

[Go to project](#)

[Preview](#)

[Edit](#)

The prompt is displayed in either *Structured* or *Freeform* view. Switch to the *Freeform* view.

Insurance claim key information extraction

Evaluate New prompt + Save work ▾

Prompt: Autosaved 1:26 PM

Structured Freeform

Read this Insurance claim description and extract the Car make and model, Location of the incident like street and date time if there is any mentioned. If you don't find these details in the description, please fill it as Not Found.

A car accident occurred on Jan 1st, 2023 at 5pm at the intersection of woodbridge. The insured vehicle, a Honda Civic, was hit by another vehicle that ran a red light. The insured driver, John, was driving within the speed limit and following all traffic rules. The accident resulted in significant damage to the insured vehicle, including a broken bumper and damaged front fender. There were no injuries reported. The insured is filing a claim for the repairs and any necessary medical expenses.

Car Details: Honda Civic;Location: Woodbridge;Date: Jan 1st, 2023;Time of Incident: 5pm

The insured vehicle, a Ford RAM, was stolen from Boston on Dec 2nd 2022. The vehicle was parked in a secure parking lot, and all necessary precautions were taken, such as locking the doors and activating the alarm system. The insured immediately reported the theft to the police and obtained a police report. The vehicle had comprehensive insurance coverage, and the insured is filing a claim for the stolen vehicle, including its estimated value, accessories, and personal belongings that were inside the vehicle at the time of theft.

Car Details: Ford RAM;Location: Boston;Date and time: Dec 2nd 2022;Time of Incident: Not Found

The insured vehicle, a Tesla model X, was vandalized on march 23rd while parked in front of the insured's residence on Magador Street. The vandalism included scratched paint, broken windows, and damage to the side mirrors. The insured promptly reported the incident to the police and obtained a police report. The insured is filing a claim for the repairs and any necessary replacement parts. The estimated cost of repairs has been assessed by a reputable auto repair shop.

Car Details: Tesla Model X;Location: Magador Street;Date: march 23rd;Time of Incident: Not Found

The insured vehicle, was parked outside during a severe hailstorm. As a result, the vehicle suffered extensive hail damage, including dents on the roof, hood, and trunk. The insured promptly reported the incident and is filing a claim for the necessary repairs.

Car Details: Not Found;Location: Parked outside;Date: Not Found;Time of Incident: Not Found

While driving on Anthony Street on 1st June, the insured vehicle, a BMW Q1, collided with a large animal (e.g., deer) that suddenly crossed the road. The accident resulted in damage to the front bumper, grille, and headlights. The insured promptly reported the incident and is filing a claim for the repairs. Additionally, the insured sought medical attention for any potential injuries resulting from the collision.

Car Details: BMW Q1;Location: Anthony Street;Date: 1st June;Time of Incident: Not Found

The insured vehicle, caught fire on April 1st due to a mechanical malfunction. The fire resulted in significant damage to the vehicle, including damage to the engine, interior, and exterior. The insured immediately contacted the fire department, and the incident was reported to the police. The insured is filing a claim for the repairs and is providing the fire department report as evidence of the fire incident.

Car Details: Not Found;Location: Not Found;Date: April 1st, 2023;Time of Incident: Not Found

(input)



In the beginning of the prompt we see the instruction for the LLM: *Read this Insurance claim description and extract the Car make and model, Location of the incident like street and date time if there is any mentioned. If you don't find these details in the description, please fill it as Not Found.*

After that we have a few examples with text and extracted entities, which means that this is a *few-shot prompt*.

At the end of the prompt we have an `{input}` variable. This variable will be passed in during inference – it will contain the text (claim summary) from which the LLM will extract entities.

In the top right corner of the *Prompt Lab* we see that this prompt was created for the *flan-ul2-20b model*.



Next to the model we have the parameters button, where we can review and add parameters for the prompt. Notice that our parameter name is *input* (referenced in the prompt as `{input}`)

Variable	Default value
input	Read this ...

Finally, we can view model parameters by clicking on the *Parameters* icon.

an-ul2-20b ▾ {#} </> X

Model parameters

Decoding

Greedy Sampling ⓘ

Repetition penalty

1 ● — 2 1

Stopping criteria ⓘ

Stop sequences

+ ↴

Min tokens

0

Max tokens

200

Enter up to 6 sequences to stop output
after the minimum number of tokens is
reached.

If you wish, test the prompt by clicking the **Generate** button. We get the results because the `{input}` variable has a default value (claim text).

{input}Car Details: Tesla Model S; Location: Parked outside; Date: April 15th, 2023

7. Return to the project view by clicking on the project name on the top of the screen.
[Projects](#) / [Insurance_LLM_Use_Cases](#) / Insurance claim key information ex [...]
8. Expand **Prompts** on the **Assets** tab. Click on the vertical ellipses menu and select **View AI factsheet**.



Projects / Insurance_LLM_Use_Cases

Overview Assets Jobs Manage

Import assets New asset +

Find assets

3 assets All assets

All assets

Name	Last modified
[...] Insurance claim key information extraction Prompt template	10 minutes ago Modified by you
[...] Insurance claim suggested next steps Prompt template	19 minutes ago Modified by Service
[...] Insurance claim summarization Prompt template	19 minutes ago Modified by Service

Evaluate View AI factsheet Promote to space View AI facts Delete

Asset types > [...] Prompts 3

Notice that the factsheet captures the details (prompt, model parameters) that you have used in the **Prompt Lab**. At this time, if you make a change in the prompt template using the **Prompt Lab**, it will be automatically reflected in the factsheet.

If you wish, in the Prompt Lab change the *max number of tokens* model parameter and verify that the changes are reflected in the factsheet.

Notice the comment on top of the factsheet: we can make changes to the prompt template until we start tracking it.

Governance



This prompt template is not tracked.

To track a prompt template, add it to an AI use case. Tracking captures details about the asset for governance purposes.

Important: Once you start tracking a prompt template in a use case, you can no longer edit it. Wait until the prompt template is stable to start tracking.

Track in AI use case



Since we made the assumption that our template is final, let's start tracking.

9. Click the **Track in AI use** case button and select the entity extraction use case you created earlier.

Insurance claim key information extraction

Track in AI use case

Track an asset to collect details about the asset in factsheets as part of a governance strategy.

Define AI use case

Choose an existing AI use case or create a new one for tracking facts about an asset

Find AI use cases

Title	Inventory
Insurance Claim Summarization	LLM Insurance Use Cases EL
Insurance Claim Entity Extraction	LLM Insurance Use Cases EL

Click **Next** on **Define Approach** screen, then select **Stable** on the **Assign model version** screen. Click **Next**.

Note: we will review approaches later in the lab.

Insurance claim key information extraction

Track in AI use case

Track an asset to collect details about the asset in factsheets as part of a governance strategy.

Assign model version

Approach: Default approach | Use case: Insurance Claim Entity Extraction

Choose the starting point for this approach.

Experimental	Stable	Custom
Use this as a starting point if your model is just starting in development and its input and output structure will likely change in the near future.	Use this as a starting point if your model is in a production state and you won't expect any major changes in its input and output structure soon.	Define your own starting version if you already tracked this model in a versioning context before.
0.0.1	1.0.0	

Version number

1.0.0

Comment (optional)

Click the **Track asset** button to finish this task.

Typically, a data scientist will perform testing prior to releasing the template for production. The *evaluation* process that we will complete in the next few steps is performed primarily for the purpose of documenting test results.



Evaluation metrics provided in watsonx are standard evaluation metrics that are used for various LLM tasks such as classification, summarization, and extractions. You can learn more about evaluation metrics in [documentation](#).

Evaluation of LLMs requires “reference data”, which is the “expected output” for the LLM. Reference data can be created manually or generated using various approaches. For generation use cases such as summarization and content generation, it’s important to provide high quality reference data, which means that it may need to be created by experts.

Example of reference data for information extraction use case:

Input Data	Reference Data
A car accident occurred on Jan 1st, 2023 at 5pm at the intersection of woodbridge. The insured vehicle, a Honda Civic, was hit by another vehicle that ran a red light. The insured driver, John, was driving within the speed limit and following all traffic rules. The accident resulted in significant damage to the insured vehicle, including a broken bumper and damaged front fender. There were no injuries reported. The insured is filing a claim for the repairs and any necessary medical expenses.	Car Details: Honda Civic; Location: Woodbridge; Date: Jan 1st, 2023; Time of Incident: 5pm

You can find reference data for our use cases in the *lab repo/Test Data* folder.

Evaluation can be done either from the factsheet view (end of the Governance tab) or the ellipses menu next to the prompt in the project view.

*******STOP HERE – A video will be provided showing the rest of the lab*******

10. Click **Evaluate** in the factsheet, then **Evaluate** on the **Run an evaluation job** screen.

 Evaluation

This prompt template is not evaluated
Evaluate your prompt template to get metrics back that measure performance and other criteria.

Evaluate



Run an evaluation job

Click Evaluate to choose dimensions to evaluate and select test data.

Evaluate

Notice that evaluation metrics relevant to our use case, *extraction*, were automatically selected.

The screenshot shows a user interface for evaluating a prompt template. On the left, there's a sidebar with 'Select dimensions', 'Select test data', and 'Review and evaluate'. The main area is titled 'Select dimensions to evaluate' with a sub-instruction: 'These dimensions are based on the prompt template task type. Learn more'. A table lists several evaluation metrics:

Dimension	Description
<input checked="" type="checkbox"/> Generative AI Quality	The Generative AI Quality monitor calculates a variety of metrics based on prompt template task type. Some metrics compare model output to the reference output you provide. Other metrics analyze model input and output and do not require reference output.
<input checked="" type="checkbox"/> ROUGE	ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. Generative text quality monitor calculates rouge1, rouge2, rougel, and rougeLSum to compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. The metric values are in the range of 0 to 1, higher value is better.
<input checked="" type="checkbox"/> Multi-label/class metrics	The Multi-label/class metrics calculate a list of quality metrics on each class.
<input checked="" type="checkbox"/> Exact match	A given predicted string's exact match score is 1 if it is the exact same as its reference string, and is 0 otherwise.

At the bottom, there are 'Cancel', 'Back', and a large blue 'Next' button.

As a reminder, the task type is captured in the factsheet.

Note: The task type is specified when we save a prompt. Since we imported a prompt, we did not need to complete this step.

The screenshot shows the 'AI Factsheet' interface. On the left, a sidebar lists 'Governance', 'Foundation model', 'Prompt template' (which is selected and highlighted in grey), 'Prompt parameters', 'Evaluation', 'Attachments' (with 'Other attachments'), and 'Other attachments'. The main panel shows the 'Prompt template' details:

- Prompt template name: Insurance claim key information extraction
- Task type: Extraction

Click **Next** on the *Evaluate Prompt Template* screen.

11. On the select data screen, navigate to the *lab repo/Test Data* folder and select the *text_extraction_claims.csv* file.

If you wish, open this file to review its content. As we discussed earlier, this file provides "expected LLM output" for the instruction that we're asking it to perform.

Select *Claims text* and the input and *Extracted Key Facts* as reference output.

Evaluate prompt template

Choose the evaluation dimensions and select the test data. [Learn more](#)

Click **Next**.

12. Click **Evaluate** to run evaluation.

Notice that the test result is *failed*. We also have 11 alerts, which means that the calculated evaluation metrics are below specified thresholds.

To view the thresholds, click on the **Configuration** icon.

The screenshot shows a dashboard titled "Generative AI Quality - Entity extraction". At the top, there's a red button labeled "Alerts triggered" with the number "11" in red. Below it, the word "Alerts" is followed by a large red "11" with a red exclamation mark inside a circle. To the right, there are buttons for "Feedback" and a dropdown menu.

Compare the *rouge* metric thresholds defined on this screen with the values from our test run.

ROUGE	
Lower thresholds	
ROUGE-1	0.8
ROUGE-2	0.8
ROUGE-L	0.8
ROUGE-Lsum	0.8

Notice that the “violation” is the difference between the threshold value and the test result value.

Metric	Score	Violation
ROUGE-1	0.75	0.05
ROUGE-2	0.72	0.08
ROUGE-L	0.75	0.05

Next, we will create another prompt template to understand prompt format that’s required for *watsonx.governance* implementation.

13. Locate the sample prompt *Extract_info_insurance_claim_llama* in the *lab repo/Prompts folder*.
14. Navigate to your project and create a new prompt. In the **Prompt Lab** use the *Freeform* view to past the sample prompt.

Make sure to select the *llama-2-70b-chat* model and set max tokens to 200.



Projects / AI gov lab / Prompt Lab

Structured Freeform

Model: llama-2-70b-chat

AI guardrails on

<><INST><>SYS>

You are a model that extracts entities from insurance claims. You specialize in finding car make and model, location, date and time of an incident.

① Read the description below and extract the car make and model, location of the incident like street and date time if there is any mentioned. If you don't find these details in the description, please fill it as Not Found. Format extracted values as a list separated by semicolon. Example: Car Details: car make and model; location; date and time. Always start response with "Car Details:" Do not include any other information. The following is an example:

Description: While driving on Anthony Street on 1st June, the insured vehicle, a BMW Q1, collided with a large animal (e.g., deer) that suddenly crossed the road. The accident resulted in damage to the front bumper, grille, and headlights. The insured promptly reported the incident and is filing a claim for the repairs. Additionally, the insured sought medical attention for any potential injuries resulting from the collision.

Car Details: BMW Q1;Location: Anthony Street;Date: 1st June;Time of Incident: Not Found

<></SYS>

Description: A car accident occurred on Jan 1st, 2023 at 5pm at the intersection of woodbridge. The insured vehicle, a Honda Civic, was hit by another vehicle that ran a red light. The insured driver, John, was driving within the speed limit and following all traffic rules. The accident resulted in significant damage to the insured vehicle, including a broken bumper and damaged front fender. There were no injuries reported. The insured is filing a claim for the repairs and any necessary medical expenses.[/INST]

Model parameters

Decoding

Greedy Sampling

Repetition penalty

1 ● 2 1

Stopping criteria ①

Stop sequences

Min tokens Max tokens

0 200

Enter up to 6 sequences to stop output after the minimum number of tokens is reached.

Reset to default

Generate →

Stop reason: End of sequence token encountered
Tokens: 406 input + 39 generated = 445 out of 4096
Time: 2.4 seconds

Notice that this prompt has been formatted specifically for *llama* (with *INST* and *SYS* tags). It's also a one-shot prompt, which, compared to the first example, will utilize less tokens for inference.

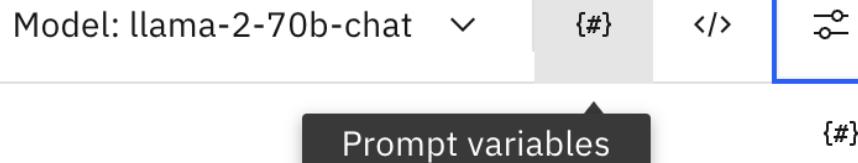
15. Click the **Generate** button to test the prompt.

Notice that the **Evaluate** button (or icon) is disabled for this prompt. In order to evaluate, track, and deploy this prompt, we need to add a parameter – the input that will be passed to the prompt.

Note: You will also not be able to track or deploy prompts without parameters.



16. In the **Prompt variables** view of the **Prompt Lab**, add the *claim_desc* parameter.



Model: llama-2-13b-chat

Prompt variables ⓘ

Variable	Default value
claim_desc	The insured ...

New variable +

Preview ⏪

You can use the following value as the default value:

A car accident occurred on Jan 31st, 2023 at 5pm at the intersection of woodbridge. The insured vehicle, a Tesla Model Y, was hit by another vehicle that ran a red light. The insured driver, John, was driving within the speed limit and following all traffic rules. The accident resulted in significant damage to the insured vehicle, including a broken bumper and damaged front fender. There were no injuries reported. The insured is filing a claim for the repairs and any necessary medical expenses.

If you wish, you can provide a different claim description from the test file we used earlier.

Next, add the parameter enclosed in curly brackets to the prompt. Notice that now the **Evaluate** button is enabled.

Projects / AI gov lab / Prompt Lab

Structured Freeform

<>>[INST] <<SYS>>

You are a model that extracts entities from insurance claims. You specialize in finding car make and model, location, date and time of an incident.

Read the description below and extract the car make and model, location of the incident like street and date time if there is any mentioned. If you don't find these details in the description, please fill it as Not Found. Format extracted values as a list separated by semicolon. Example: Car Details: car make and model; location; date and time. Always start response with "Car Details:" Do not include any other information. The following is an example:

Description: While driving on Anthony Street on 1st June, the insured vehicle, a BMW Q1, collided with a large animal (e.g., deer) that suddenly crossed the road. The accident resulted in damage to the front bumper, grille, and headlights. The insured promptly reported the incident and is filing a claim for the repairs. Additionally, the insured sought medical attention for any potential injuries resulting from the collision.

Car Details: BMW Q1;Location: Anthony Street;Date: 1st June;Time of Incident: Not Found

<</SYS>>

Description: [claim_desc][INST]

Prompt variables ⓘ

Variable	Default value
claim_desc	A car accident ...

Preview ⏪

Generate →

- Save the prompt template by clicking the save icon then save as. Make sure to select Extraction as the task type.

Unsaved

New prompt +

Save as

Model: llama-2-70b-chat

Important Note: At this time the task type will automatically determine which evaluation tests will be run. You will not be able to change the task type after saving the template.

Save your work

X

Specify how to save your work by selecting an asset type and defining details.

Asset type

Prompt template

Save the current prompt only, without its history.

Notebook

Save the current prompt as a notebook.

Prompt session

Save history and data from the current session.

Define details

Name

Task

Description (optional)
What's the purpose of this prompt asset?

View in project after saving ⓘ



18. Click the evaluate icon to run evaluation for this prompt. Notice that the *rouge* score for this prompt is higher, it does not breach the configured threshold.

AI Factsheet Evaluate

Generative AI Quality - Entity extraction

Alerts triggered

Alerts 8

Feedback

Metric	Score	Violation
ROUGE-1	0.85	none
ROUGE-2	0.71	0.09
ROUGE-L	0.84	none
ROUGE-Lsum	0.84	none

Next, we will enable tracking for this template following the same steps as you've done for the other prompt.

Before we do that, we will create a new *approach* for our use case. Approaches help us organize different *implementations* for the same use case. For example, some companies may do A/B testing to find out which prompt/model deliver the best performance.

19. Navigate to the *AI use cases* view and select your use case.

AI use cases

Find a AI use case

Name	Status	Owner	Inventory	Tags
Insurance Claim Entity Extraction	Development in progress	EL Elena Lowery	LLM Insurance Use Cases EL	llm entity extraction

20. In the **Lifecycle** view click **New Approach**.

Welcome to your new AI use case

Discover next steps

Organize team work with approaches

Manage AI assets with versions

Show deleted assets

New approach +

Default approach

Latest: 1.0.0 | 1 version

A default approach for tracking your AI assets.

21. Create the new approach using values similar to the ones shown in the screenshot and click **Create**.

New approach

An approach defines one path for solving the goal of the use case. For example, an approach might be a variation on a machine learning model, or a challenger model. Each approach can include multiple versions.

Icon: Packages

Color: Gray

Title: One shot extraction with llama

Description (optional): Prompt that uses one-shot example and llama 70B model

Create

At this time we do not have any assets tracked for this approach.

Show deleted assets

One shot extraction with llama

Prompt that uses one-shot example and llama 70B model

No AI assets tracked in this approach.

22. Navigate to the **Factsheet** view of your *llama* prompt (from the project **Assets** view).

Name	Last modified
[...] Info_extraction_llama_70B	Now Modified by Service
[...] Insurance claim key information extraction	37 minutes ago Modified by you
[...] Insurance claim summarization	59 minutes ago Modified by Service
[...] Insurance claim suggested next steps	59 minutes ago Modified by Service

23. Click **Track in AI use case**. Select your use case and the newly created approach.

This prompt template is not tracked.
To track a prompt template, add it to an AI use case. Tracking captures details about the asset for governance purposes.
Important: Once you start tracking a prompt template in a use case, you can no longer edit it. Wait until the prompt template is stable to start tracking.

[Track in AI use case](#)

Info_extraction_llama_70B

Track in AI use case

Track an asset to collect details about the asset in factsheets as part of a governance strategy.

Define AI use case

Define approach

Assign model versions

Define approach

Use case: **Insurance Claim Entity Extraction**

An approach defines one path for solving the goal of the use case. Ex. an approach might be a variation of a prompt template. Each approach can include multiple versions.

New approach +

<p>One shot extraction with llama</p> <p>Prompt that uses one-shot example and llama 70B model</p>	<p>Default approach</p> <p>A default approach for tracking your AI assets.</p>
--	--

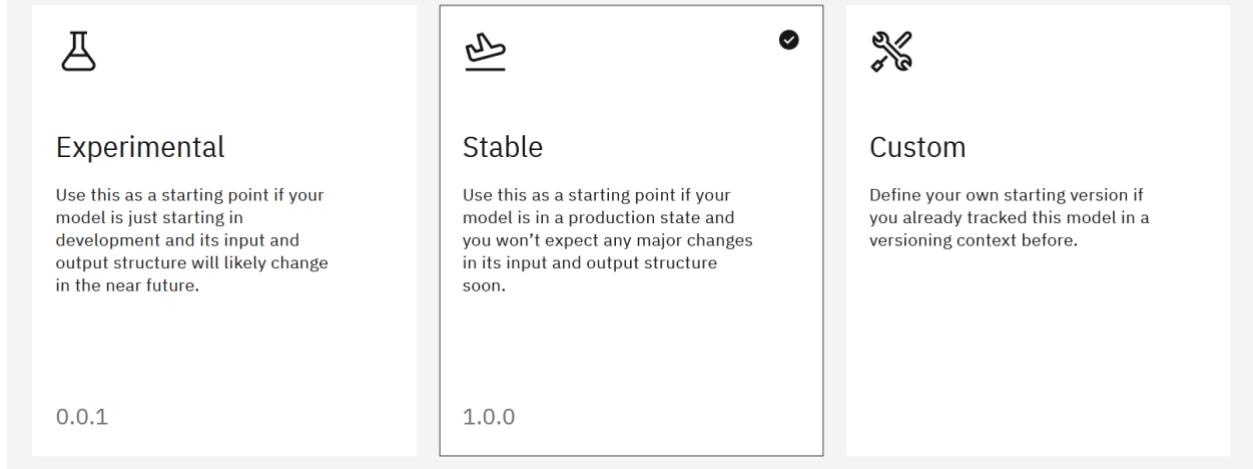
Latest: 1.0.0 | 1 Version

Select the *Stable* model version, click **Next**, then click **Track asset**.

Assign model version

Approach: **One shot extraction with llama** | Use case: **Insurance Claim Entity Extraction**

Choose the starting point for this approach.



Version Type	Description	Status
Experimental	Use this as a starting point if your model is just starting in development and its input and output structure will likely change in the near future.	0.0.1
Stable	Use this as a starting point if your model is in a production state and you won't expect any major changes in its input and output structure soon.	1.0.0
Custom	Define your own starting version if you already tracked this model in a versioning context before.	

If you wish, come back to the **AI use case** view and verify that both templates are now tracked – one under the *Default* approach, and one under the custom one that we created.

Next, we will promote both prompt templates to production. In this lab we will skip the validation step. Validation is similar to model evaluation that we have done in this lab, but it's usually done by a different team in a different *watsonx* project. If you would like to learn more about the validation step, you can review [documentation](#).

24. From the main menu navigate to **Deployments** and create a new deployment space using values similar to the following screenshot.

Notes:

- Make your deployment space name unique by adding your initials.
- Your machine learning service name will be different.

IBM watsonx

Filter navigation

- Home
- Data
 - Platform connections
- Projects
 - View all projects
 - Jobs
- AI governance
 - AI use cases
 - External models
- Deployments**
- Samples
- Documentation
- Administration

IBM watsonx

Upgrade 3 EL ...

Depployments

4 spaces

New deployment space +

Activity Spaces

Create a deployment space

Use a space to collect assets in one place to create, run, and manage deployments

Define details

Name: Insurance Use Cases EL - Production

Description (Optional)

Deployment space description

Select services

Select storage service: Cloud Object Storage-ur

Select machine learning service (optional): Machine Learning-fz

Upload space assets (optional)

Populate your space with assets exported from a project or space to a .zip file. You can add more assets after the space is created.

Drop .zip file here or browse your files to upload

Deployment stage: Production

Deployment space tags (optional)

Add a tag

Cancel Create

25. Navigate back to your projects. From the **Assets** view of the project, select **Promote to Space**.

26. Select the space that you previously created and click **Promote**.

All assets

Name	Last modified	
[...] Info_extraction_llama_70B ⓘ Prompt template	9 minutes ago Modified by Service	<input type="button" value="Evaluate"/> <input type="button" value="View AI factsheet"/> <input style="border: 2px solid #0070C0; border-radius: 5px; padding: 2px 10px;" type="button" value="Promote to space"/>
[...] Insurance claim key information extraction ⓘ Prompt template	46 minutes ago Modified by you	<input type="button" value="Evaluate"/> <input type="button" value="View AI factsheet"/>
[...] Insurance claim summarization ⓘ Prompt template	1 hour ago Modified by Service	<input type="button" value="Evaluate"/> <input type="button" value="View AI factsheet"/>
[...] Insurance claim suggested next steps	1 hour ago	<input type="button" value="Evaluate"/> <input type="button" value="View AI factsheet"/>

Promote to space

Use a deployment space to organize supporting resources such as input data and environments; deploy models or functions to generate predictions or solutions; and view or edit deployment details.

Target space

x v

Why don't I see all of my spaces? [\(i\)](#)

Go to the space after promoting the prompt template

Selected assets (1)

Name	Format
Info_extraction_llama_70B	Prompt template

Select version

[**i**](#) Promoting a version of an asset to a space creates a new asset in the space, with a new asset ID.

27. Navigate to the **Deployment space**. Next, we will deploy both templates.

In the **Assets** view of the space, click on **Deploy** next to the template.

Insurance Claims EL - production

Overview **Assets** Deployments Jobs Manage

Find assets Import assets

2 assets All assets (2)

Asset types: Info_extraction_llama_70B

Name	Last modified
[...] Info_extraction_llama_70B	5 minutes ago
Prompt template	Service

Deploy Delete ...

Provide a deployment name and click **Create**.

Create a deployment

Deployment type

Online

Run the prompt template on data in real-time, as data is received by a web service.

Name: Extract_claim_info_llama

Serving name: Deployment serving name

Description: Deployment description

Create

28. Click on the created deployment and explore the various tabs.

Deployments / Insurance Claims EL - production / Info_extraction_llama_70B /

Extract_claim_info_llama Deployed Online

API reference Test Evaluations AI Factsheet

Direct link

Private endpoint

Text endpoint: <https://private.us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/eeaceab3-4d4c-4f3b-833a-0a2a2a2a2a2a>

Stream endpoint: <https://private.us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/eeaceab3-4d4c-4f3b-833a-0a2a2a2a2a2a>

Public endpoint

Text endpoint: <https://us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/eeaceab3-4d4c-4f3b-833a-0a2a2a2a2a2a>

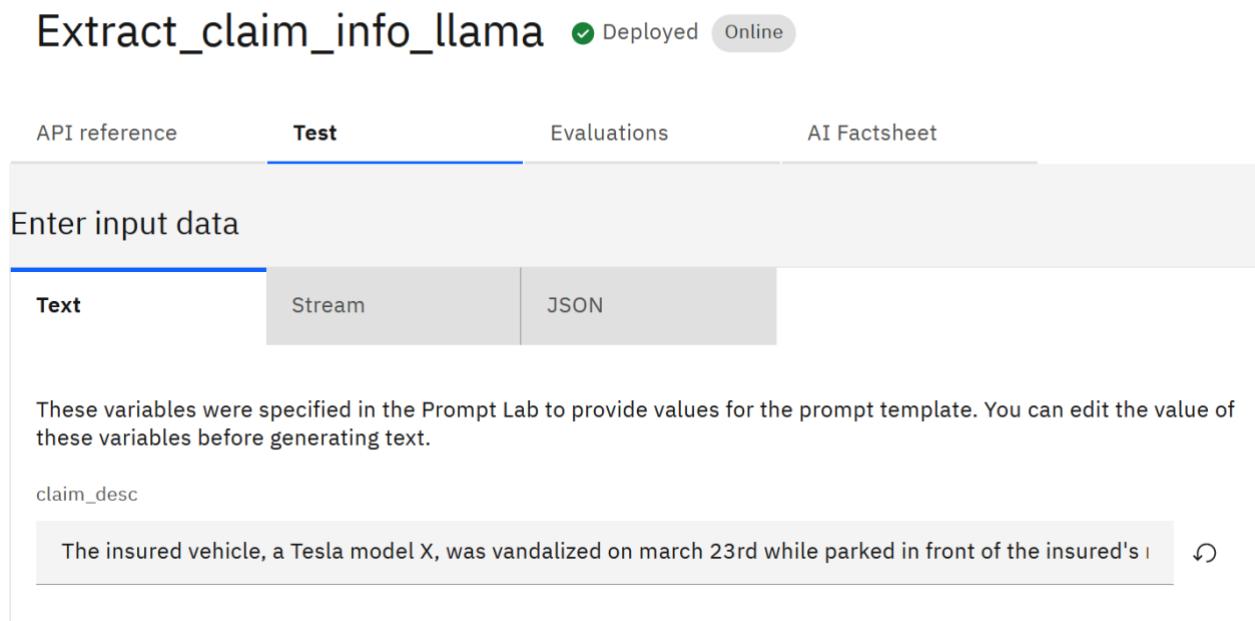
Stream endpoint: <https://us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/eeaceab3-4d4c-4f3b-833a-0a2a2a2a2a2a>

[Learn more](#) about the 2021-05-01 version query parameter

On the **Test** tab we can interactively test the prompt.

If you wish, you can copy one of the claim description values from the provided csv file that was used for evaluations, and test the LLM.

[Deployments](#) / [Insurance Claims EL - production](#) / [Info_extraction_llama_70B](#) /



The screenshot shows the deployment interface for 'Extract_claim_info_llama'. At the top, there's a navigation bar with 'Deployments' and 'Insurance Claims EL - production' selected. Below it, the deployment name 'Extract_claim_info_llama' is shown with a green checkmark indicating it's 'Deployed' and 'Online'. There are tabs for 'API reference', 'Test' (which is currently selected), 'Evaluations', and 'AI Factsheet'. Under the 'Test' tab, there's a section for 'Enter input data' with three options: 'Text' (selected), 'Stream', and 'JSON'. Below this, a note says: 'These variables were specified in the Prompt Lab to provide values for the prompt template. You can edit the value of these variables before generating text.' A variable 'claim_desc' is listed with the value: 'The insured vehicle, a Tesla model X, was vandalized on march 23rd while parked in front of the insured's ...'. There's also a small icon with a circular arrow.

The **AI Factsheet** tab shows the same information as the factsheet in the project.

Next we will walk through the process of setting up *Evaluations*.

The process of evaluation, whether it's done manually in a project, manually in the *Deployment Space*, or automatically is always the same: generated output is compared with the reference data. The only difference is how generated and reference data are provided.

In Deployment Spaces, we can set up automatic payload logging (payload data is the data that's passed in to the LLM). You can find more information about setting up payload logging in [documentation](#).

Payload logging is not in the scope of this lab. Since we do not have payload logging configured, we can go through the setup process and run manual evaluation.

29. On the *Evaluations* tab click **Activate**.

Extract_claim_info_llama Deployed Online

[API reference](#) [Test](#) [Evaluations](#) [AI Factsheet](#)



Activate monitoring

Click Activate to choose dimensions to evaluate.

[Activate](#)

30. Click on **Next** on the **Select dimensions** screen (accept defaults).

Evaluate prompt template

Choose the evaluation dimensions and select the test data. [Learn more](#)

<input checked="" type="radio"/> Select dimensions <input type="radio"/> Select test data <input type="radio"/> Review and evaluate	Select dimensions to evaluate <small>These dimensions are based on the prompt template task type. Learn more</small> <table border="1" style="width: 100%;"> <thead> <tr> <th style="background-color: #f2f2f2;">Dimension</th> <th style="background-color: #f2f2f2;">Description</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> Generative AI Quality</td> <td>The Generative AI Quality monitor calculates a variety of metrics b reference output you provide. Other metrics analyze model input a</td> </tr> </tbody> </table>	Dimension	Description	<input checked="" type="checkbox"/> Generative AI Quality	The Generative AI Quality monitor calculates a variety of metrics b reference output you provide. Other metrics analyze model input a
Dimension	Description				
<input checked="" type="checkbox"/> Generative AI Quality	The Generative AI Quality monitor calculates a variety of metrics b reference output you provide. Other metrics analyze model input a				

31. On the **Select test data** screen browse to the *lab repo/Test Data folder* and select the *test_extraction_claims.csv* file.

Evaluate prompt template

Choose the evaluation dimensions and select the test data. [Learn more](#)

<input checked="" type="radio"/> Select dimensions <input type="radio"/> Select test data <input type="radio"/> Review and evaluate	 Drop a file here or browse for a file to upload <small>Add a CSV file that include input and output examples. Maximum s records is 10.</small> <p>Browse</p>
---	---

32. Similar to the evaluation configuration in the project, select the *Input* and *Reference output* fields as showing the following screenshot.

Evaluate prompt template

Choose the evaluation dimensions and select the test data. [Learn more](#)

The screenshot shows the 'Evaluate prompt template' interface. On the left, a sidebar lists three steps: 'Select dimensions' (checked), 'Select test data' (unchecked), and 'Review and evaluate' (unchecked). The main area is titled 'Map prompt variables to columns' with the sub-instruction 'For each prompt variable, select the associated column.' Below this is a 'Field separation' section with a dropdown set to 'Comma (,)'. Under 'Input', there is a field labeled 'Claims text'. Under 'Reference output', there is a field labeled 'Extracted Key Facts'.

33. Click **Next**, then click **Evaluate**.

Evaluate prompt template

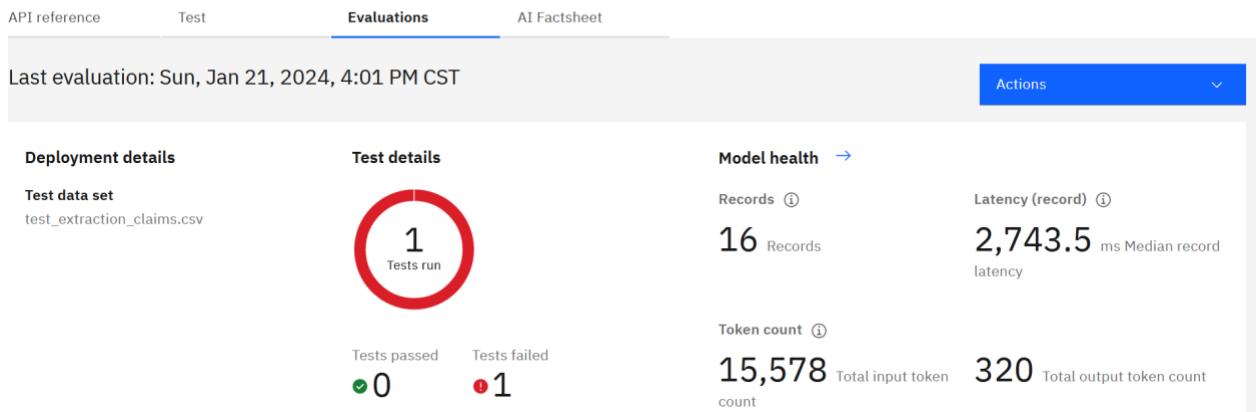
Choose the evaluation dimensions and select the test data. [Learn more](#)

The screenshot shows the 'Evaluate prompt template' interface. On the left, the sidebar shows 'Select dimensions' (checked), 'Select test data' (checked), and 'Review and evaluate' (unchecked). The main area is titled 'Review' and contains the following settings: 'Task: Entity extraction', 'Test data: test_extraction_claims.csv', and 'Evaluations: Generative AI Quality'. A note on the right states: 'Evaluation can take a few minutes to complete. You can continue to work on other things while your evaluation is in progress.'

Review evaluation results.



Get_key_info_prompt Deployed Online



Next, we will invoke the deployed template from a client application. If you would like to do a quick test, you can use a notebook that's running in *watsonx*. However, in production deployment, the client application will be running outside of *watsonx*. We will explain where to find and how to modify the REST call for invoking the template, and you can decide how you would like to test it.

34. In your *Deployment Space* project click on the **API reference** tab.

Deployments / Insurance claims - production / Insurance claim key information ... /

Get_key_info_prompt Deployed Online

API reference Test Evaluations AI Factsheet

Direct link

Private endpoint Stream endpoint Bearer <token>

Text endpoint https://private.us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/23c8e34f-44ea-4ca9-b2... https://private.us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/23c8e34f-44ea-4ca9-b2...

Public endpoint Stream endpoint IAM

Text endpoint https://us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/23c8e34f-44ea-4ca9-b2... https://us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/23c8e34f-44ea-4ca9-b2...

Learn more about the 2021-05-01 version query parameter

Code snippets

cURL

```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account (https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ml-authentication.html)
curl --insecure -X POST --header "Content-Type: application/x-www-form-urlencoded" --header "Accept: \napplication/json" --data-urlencode "grant_type=urn:ibm:params:oauth:grant-type:apikey" \
--data-urlencode "apikey=$API_KEY" "https://iam.cloud.ibm.com/identity/token"

# the above CURL request will return an auth token that you will use as $IAM_TOKEN in the scoring request below
# TODO: manually define and pass values to be scored below
curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" --header "Authorization: \nBearer $IAM_TOKEN" -d '{"parameters": [{"prompt_variables": {"input": "Read this Insurance claim description and extract the Car make and model, Location of the incident like street and date time if there is any mentioned."}}]}
```

If you would like to do a quick test with a notebook, complete the next two steps. You can also skip these steps and go the step of testing prompt template invocation with a Python script.

Note: if you typically don't work with the REST API, we recommend that you skip the next 2 steps.

35. Create a new notebook in your watsonx project (open a new browser tab). Use the default Python environment when creating the notebook.

36. Copy the code from the *Code snippets* into a new notebook.

After the initial copy, break up the code for getting the token and invoking the prompt into 2 cells. We need to do this to get the token value that we will paste into the invocation cURL command.

Update the first cell with your API key, run the cell, then copy the token into the cURL command in the 2nd cell.

Note that the sample code in deployment points to the streaming text URL. If you wish, you can replace it with the text endpoint URL, which you can find on the same **API reference** tab.

```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account (https://datapl...  
!curl --insecure -X POST --header "Content-Type: application/x-www-form-urlencoded" --header "Accept: application/json" --data-urlencode "grant_type=urn:ibm:params:oauth:grant-type:apikey" \--data-urlencode "apikey=*****" "https://iam.cloud.ibm.com/identity/token"
```

{"access_token": "eyJraWQi0iIyMDI0MDEwNjA4Mzc1LCJhbGci0iJSUzI1NIJ9.eyJpYW1faWQj0iJJQk1pZC01MEFW0QNNRzBQIiwWlkIiwanRpIjoiNjk1Mjk0MDItMzFimS000Tk5LTk3NjktYjE5N2J1NGVjMGNKIiwiwRlbnRpZmlciI6ijUwQVZTQ01HMFAiLCJna>Yw8iLCJuYW1lIjoiQ2F0aGVyaW5lIENhb1sImVtYWLsIjoiY2F0aGVyaW5lLmNhb0BpYm0uY29tIiwiC3ViIjoiY2F0aGVyaW5lLmNhhtb0BpYm0uY29tIiwiwFtX2lkIjoiSUJNaWQtNTBBV1NDTUcwUCIsIm5hbWUi0iJDYXRoZXJpbmUgQ2FViIwiZ2l2ZW5fbmFtZSI6IkNfJjYXRoZXJpbmUuY2FvOGLibS5jb20ifSwjYWNjb3VudCI6eyJ2YxpZCI6dHJ1ZSwiYnNzIjoiZDjjMmIwYzkwYmFjNDk2ZmEwNmUyJknJvemVuIjp0cnVLLCJpbXMi0iIyNjEzM2Y0In0sImlhcdI6MTcwNjc0MDIyNiwiZXhwIjoxNzA2NzQzODI2LCJpc3Mi0iJodHRwcovL2ZSI6InVyb1ppYm06cGFyYW1z0m9hdXRo0mdyYW50LXR5cGU6YXBpa2V5IiwiC2NvcGUi0iJpYm0qb3BlbkIiwiY2xpZW50X2lkIjoiZ

Test invoking the prompt with the single text output

```
(*) # The cURL command in the cell above above CURL request will return an auth token that you will use as $IAM_TOKEN in the scoring request b  
# Include the token without quotes or any other characters, for example Bearer abc123  
# Replace the last line of  
  
# Note this code is testing the private URL with text generation (last line of the request), and not streaming  
!curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" --header "Authorization: Bearer $IAM_TOKEN" \  
-d '{ "parameters": { "prompt_variables": { "input": "The insured vehicle, a Tesla model X, was vandalized on March 23rd while \n\nparked in front of the insured residence on Magador Street. The vandalism included scratched paint, broken windows, and damage to the side door. The insured promptly reported the incident to the police and obtained a police report. The insured is filing a claim for the repairs and an replacement parts. The estimated cost of repairs has been assessed by a reputable auto repair shop." } } }' \  
"https://private.us-south.ml.cloud.ibm.com/ml/v1-beta/deployments/23c8e34f-44ea-4ca9-b256-8b09865b62f5/generation/text?version=2021-05-01"
```

Cell 2

In the next step you will run a Python client script that invokes the template. This code invokes the same deployed template as the notebook, but we refactored the code to make it easier to understand and maintain.

37. Find the following Python scripts in the downloaded lab repo /Scripts folder:

- *demo_invoke_template.py*

Let's update and review the script.

On top of the script replace variables with your IBM cloud API key and the public URL of your deployed template.



```
import requests, json

# Replace with your IBM Cloud API key
cloud_api_key = ''

# In most cases the URL for authentication should be this value.
# If you get an authentication error, check the URL in IBM Cloud
auth_url = 'https://iam.cloud.ibm.com/identity/token'
# Make sure to provide public, text URL (not private and not straming)
prompt_url = ''
```

The script has the following functions:

- *get_credentials()*: generates the authentication token
- *invoke_prompt()*: invokes the prompt
- *demo_invoke_prompt()*: invokes all other functions for testing

38. Run the script. The output will be shown in Python terminal.

```
C:\ProgramData\anaconda3\envs\Python310\python.exe C:\Users\1A3030897\PycharmProjects\LLM_Workshop\src\main\python\prompt.py
The access token is: eyJraWQiOiIyMDI0MDEwNjA4MzciLCJhbGciOiJSUzI1NiJ9.eyJpYW1faWQiOiJJQk1pZC0xMTA
The generated text is: Car Details: Tesla Model X;Location: Magador Street;Date:
Process finished with exit code 0
```

You have finished testing the deployed prompt template.

Conclusion

You have finished the **AI Governance in watsonx** lab. In this lab you learned:

- Best practices for organizing AI use cases
- Tracking prompt template lifecycle in watsonx
- Invoking the deployed prompt template from a client application.