

# IBM watsonx.ai Technical Essentials

Technical Hands-on Lab

Course code W7S169G

**Andre de Waal** ([andre.dewaal@ibm.com](mailto:andre.dewaal@ibm.com))  
Learning Content Development, Global Sales

This document contains instructions for all the guided exercises presented in the ‘IBM watsonx.ai Technical Essentials’ course. Use it for future reference or during the course workflow as guidance to perform the tasks.

To access the classroom environment, you log in to the IBM Cloud with your valid IBM Cloud credentials. You use the IBM’s Software as a Service (SaaS) offering available in the IBM public cloud. If you do not currently have valid IBM Cloud credentials, the first exercise guides you to register for an IBM Cloud account to obtain your IBM Cloud credentials.

The rest of the prompt engineering exercises are done in the IBM public cloud by using the watsonx.ai platform.

## Contents

1. Prepare the Lab Environment .....	3
1.1 Create an IBM Cloud account.....	3
1.2 Log in to watsonx .....	6
1.3 Setting Up the Lab Environment .....	10
2. Prompt engineering .....	23
2.1 Writing your first prompt .....	23
Section 1. Zero-shot prompting.....	24
Section 2: Single-shot and two-shot prompting .....	33
Section 3: Inspect the foundation model Python code .....	45
Section 4: Model options and tokens.....	51
3. Exercise review and wrap-up .....	57

# 1. Prepare the Lab Environment

## Estimated time

20:00 minutes (to complete this section)

## Overview

In this exercise, you set up your IBM Cloud account and log in to the watsonx platform. If you already have an account, you can skip the creation of an IBM Cloud account (Section 1.1 of this lab guide).

After the creation of an IBM Cloud account, you switch platforms and log in to the watsonx platform.

## Objectives

After completing this exercise, you should have an existing IBM Cloud account and be logged in to watsonx.

To complete all exercises in this lab, you need to have a running instance of Watson Machine Learning. You create a Watson Machine Learning service by using the Lite plan and associate the service to a project.

At the end of this exercise, you have access to the wasonx.ai Prompt Lab.

## Requirements

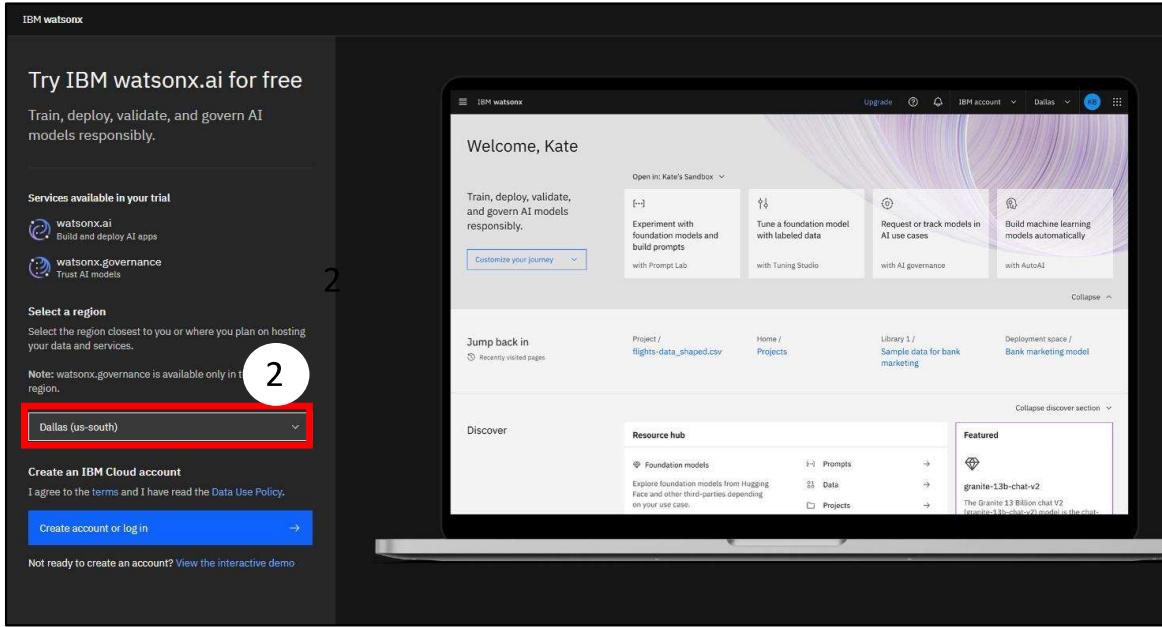
- A valid email address and password.
- Personal information such as name, country, and phone number.
- Billing and payment information.

### 1.1 Create an IBM Cloud account

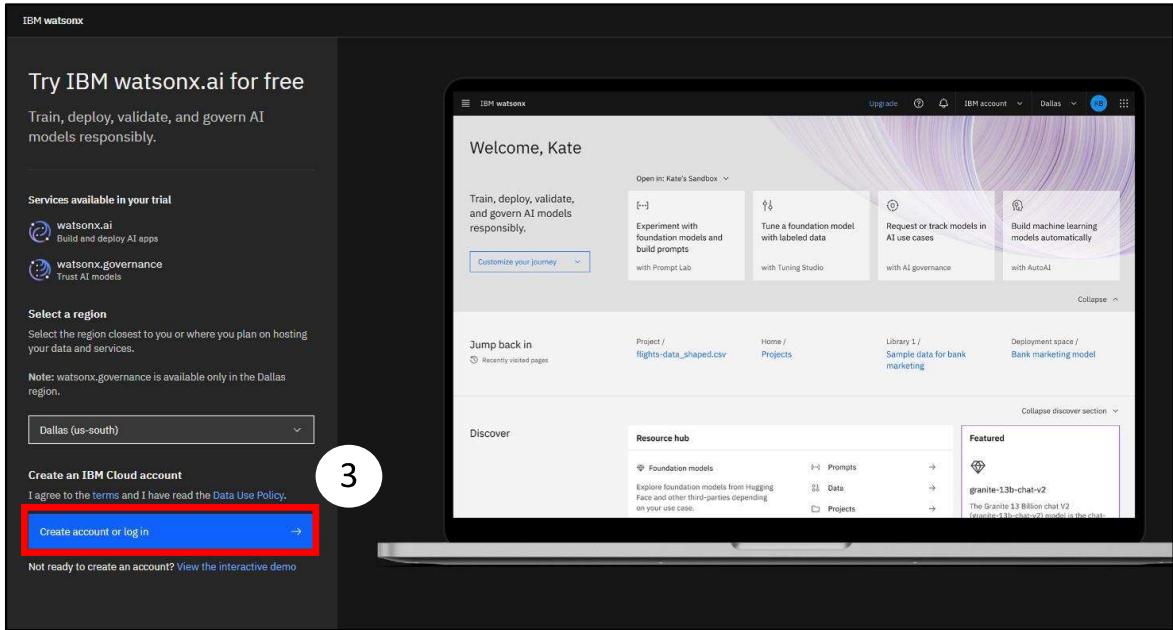
To get started, you need to get access to the lab environment in IBM Cloud.

1. If you currently have valid credentials to an IBM Cloud account, proceed to section 1.2 (Log in to watsonx). If you **do not** currently have valid IBM Cloud credentials, navigate to [Try IBM watsonx.ai for free | IBM watsonx](#) page and complete the registration process by following the steps in this section.

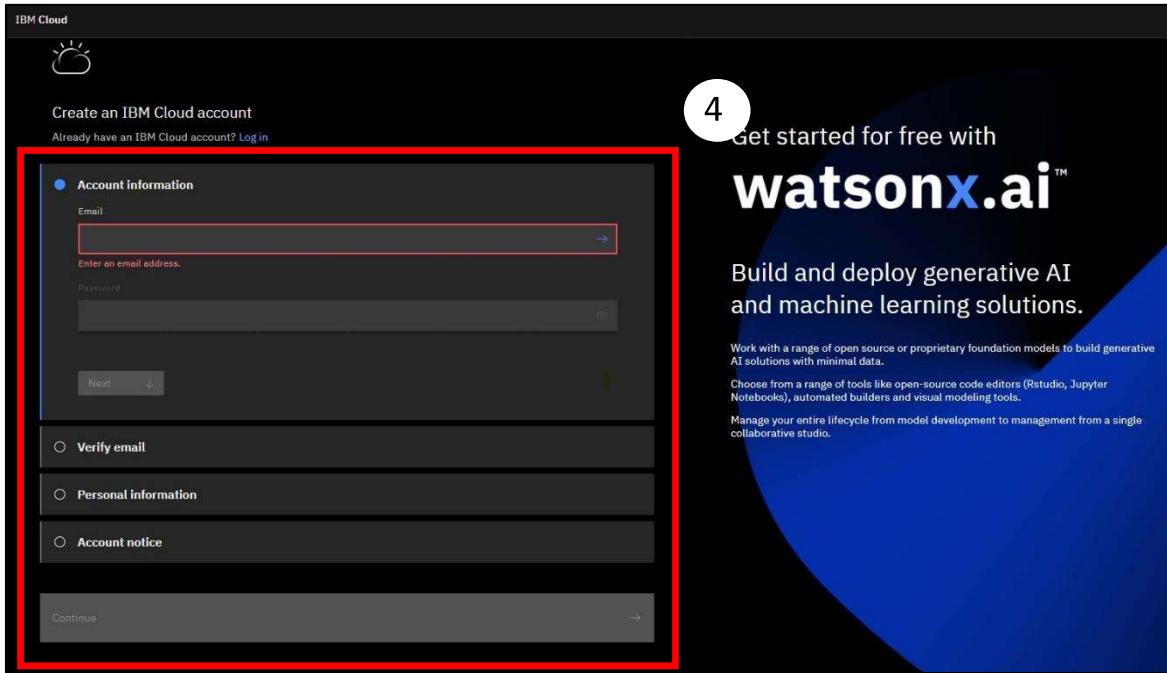
- Click on the **Select a region** drop-down menu and select the **Dallas (us-south)** region.



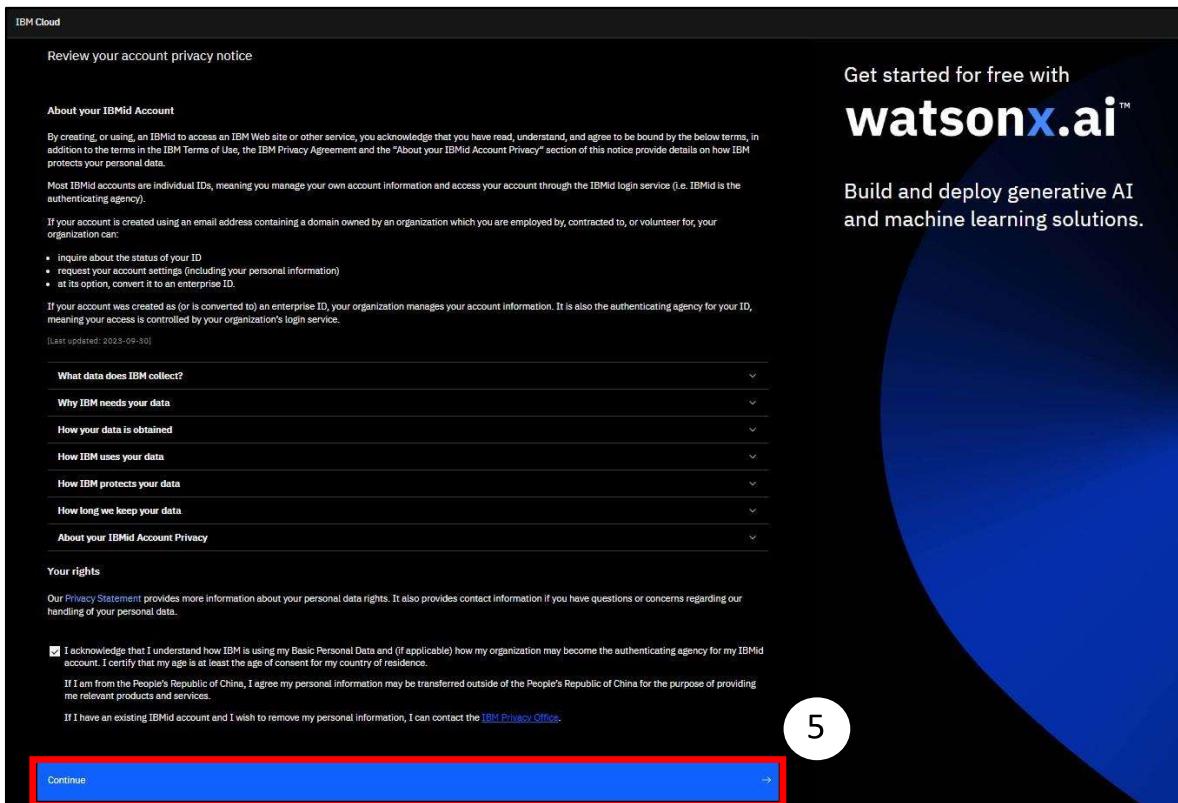
- Click the **Create account or log in** button.



- If you are not a current IBM Cloud account holder, complete the account information, verify your e-mail, accept the product terms and conditions, and click **Continue**. If you are an IBM Cloud account holder, log in to verify that your account is still active and continue with Section 1.2.

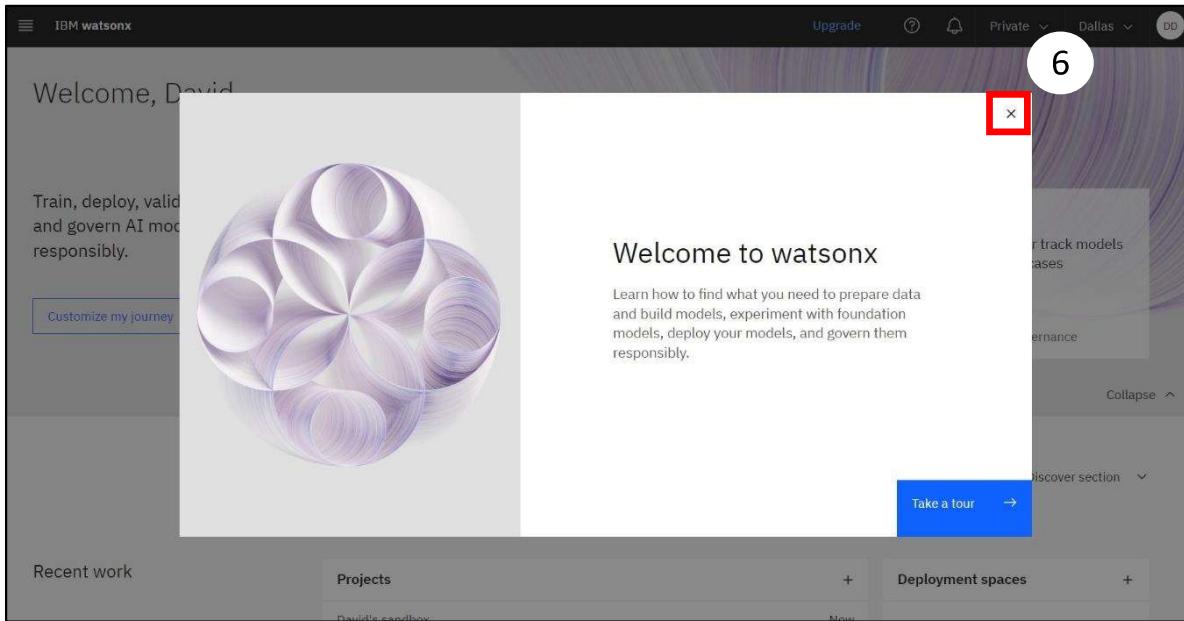


- Review how IBM will use your personal data and click **Continue** (remember to record your **IBM Cloud credentials** for future use).



- Select the **I Agree to the terms and data use policy** check box and close the Welcome to

watsonx window by clicking on the **x** in the upper-right corner of the window.



You successfully registered and logged in to **watsonx**.

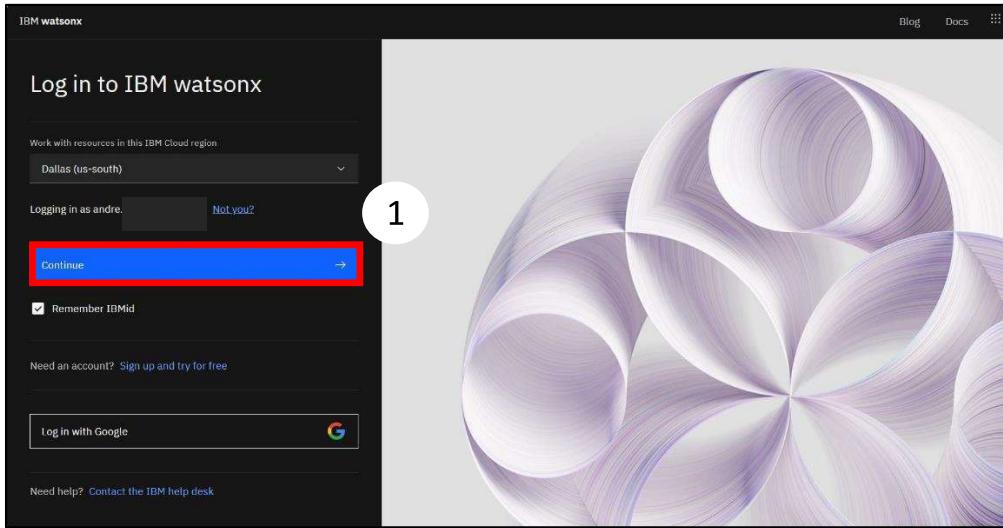
7. Continue with Section 1.3.

## 1.2 Log in to watsonx

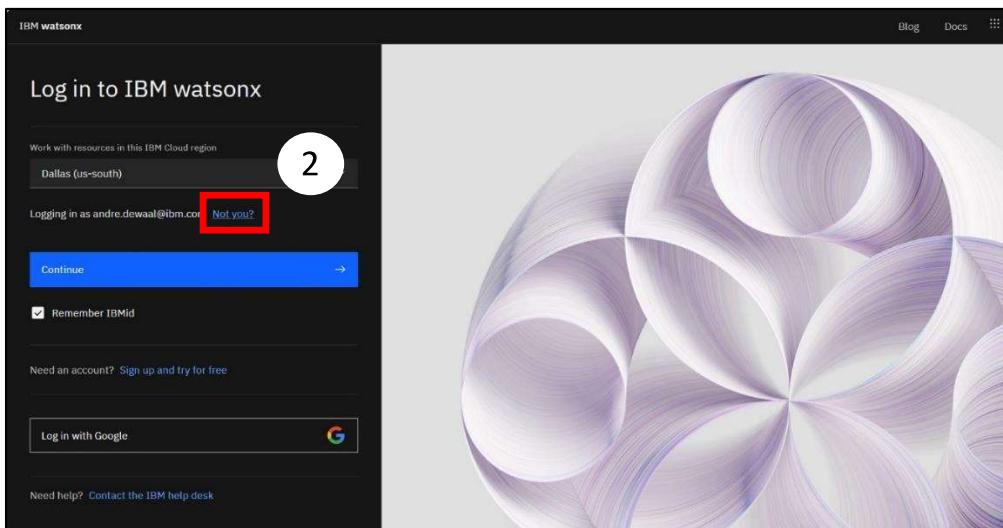
To get access to the lab environment, you log in to the watsonx platform.

1. Go to the [IBM watsonx](#) login page and log in with your **IBMid** and **password**. If the displayed credentials are correct, click **Continue** to log in to the watsonx platform and then proceed with step 4. Otherwise, continue with step 2.

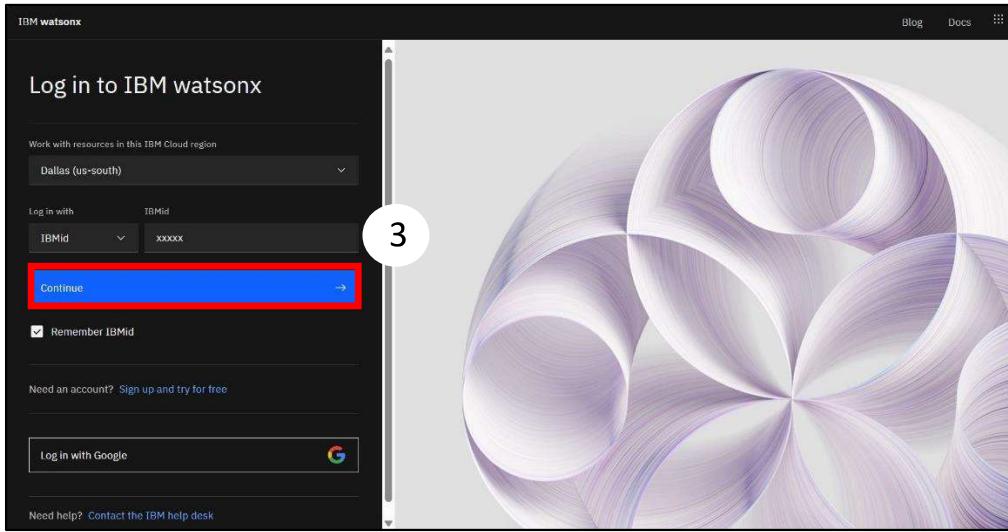
Note: The number **x** in the black (or white) circle corresponds with the action that you need to perform in step **x** of this section of the lab. You are currently on step 1 of this section.



2. Click **Not you?** to take you to a log in screen where you can provide your IBM Cloud credentials to log in to watsonx.

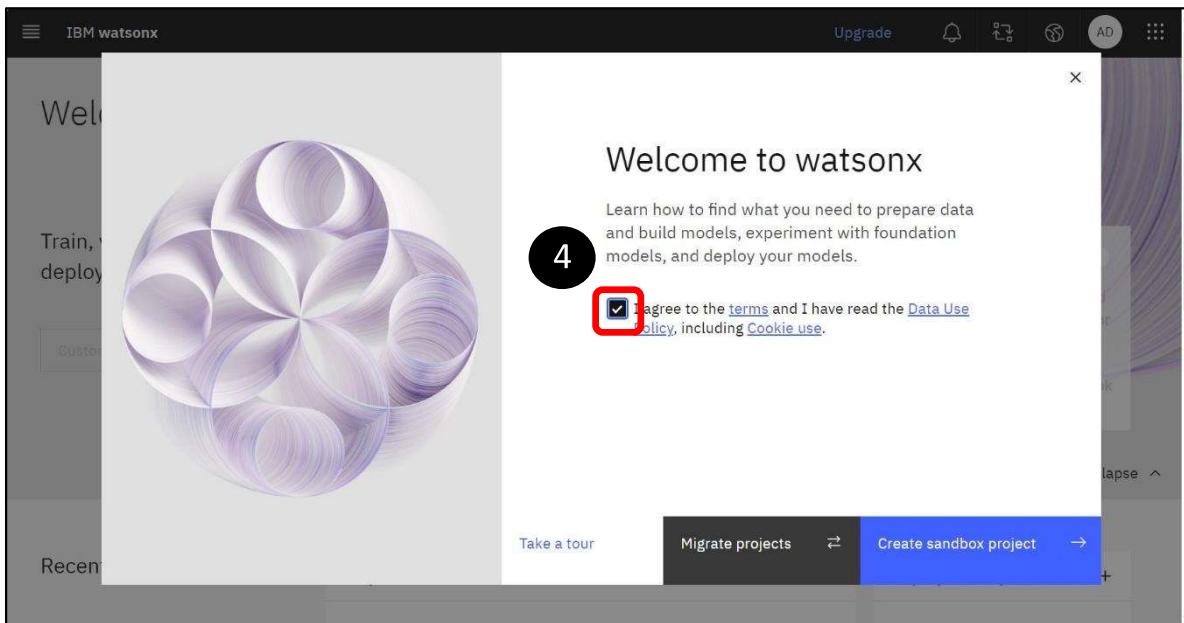


3. Fill in your IBM Cloud id in the **IBMid** text box (replace xxxx with your Cloud id) and click **Continue** (provide your password if required).

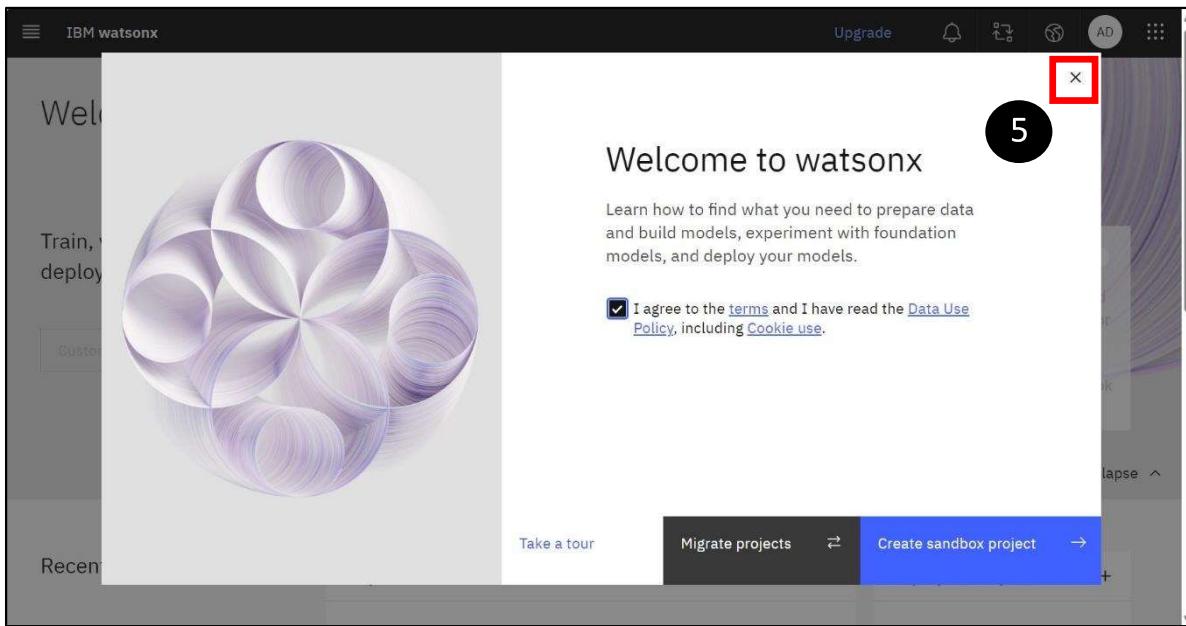


4. If this is the first time that you are logging in to the IBM Watsonx platform, continue with this step. Otherwise, continue with step 7.

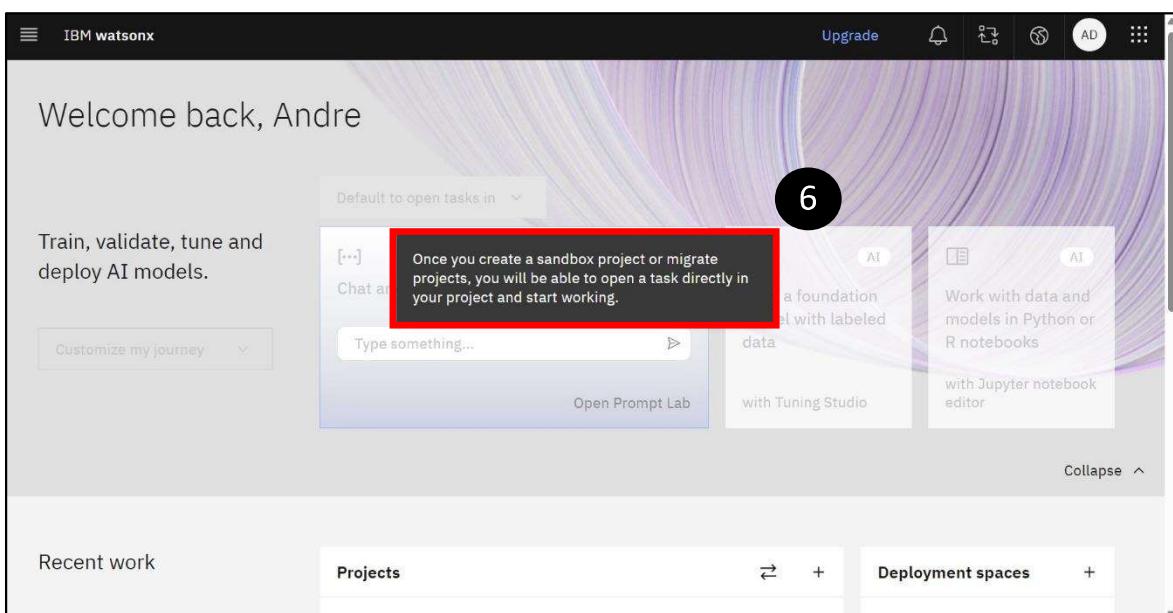
You are greeted with a “Welcome to Watsonx” message. Select the Terms checkbox.



5. Click the x in the upper-right quadrant of the page to close the Welcome to Watsonx message.



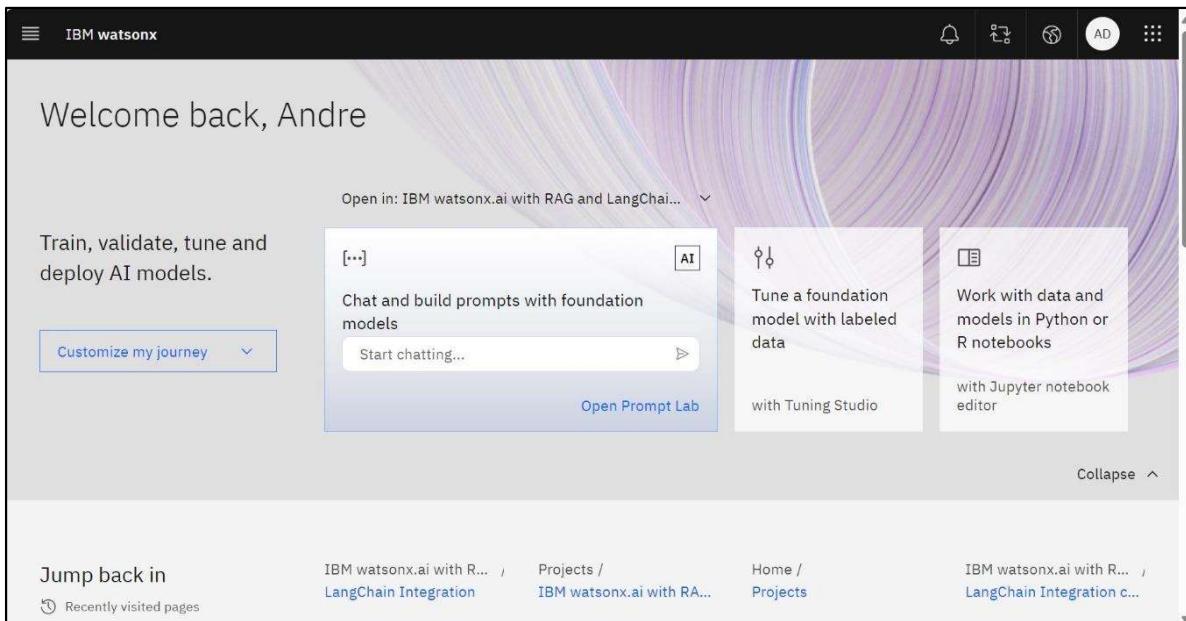
6. You are greeted by the **IBM watsonx** home page.



**Note:** You only get the warning message in the middle of the screen if you have never experimented with the Prompt Lab in watsonx. You create a project in section 1.2 that satisfies this requirement.

You successfully logged into the IBM watsonx platform. Continue with Section 1.3.

7. You are greeted by the **IBM watsonx** home page.



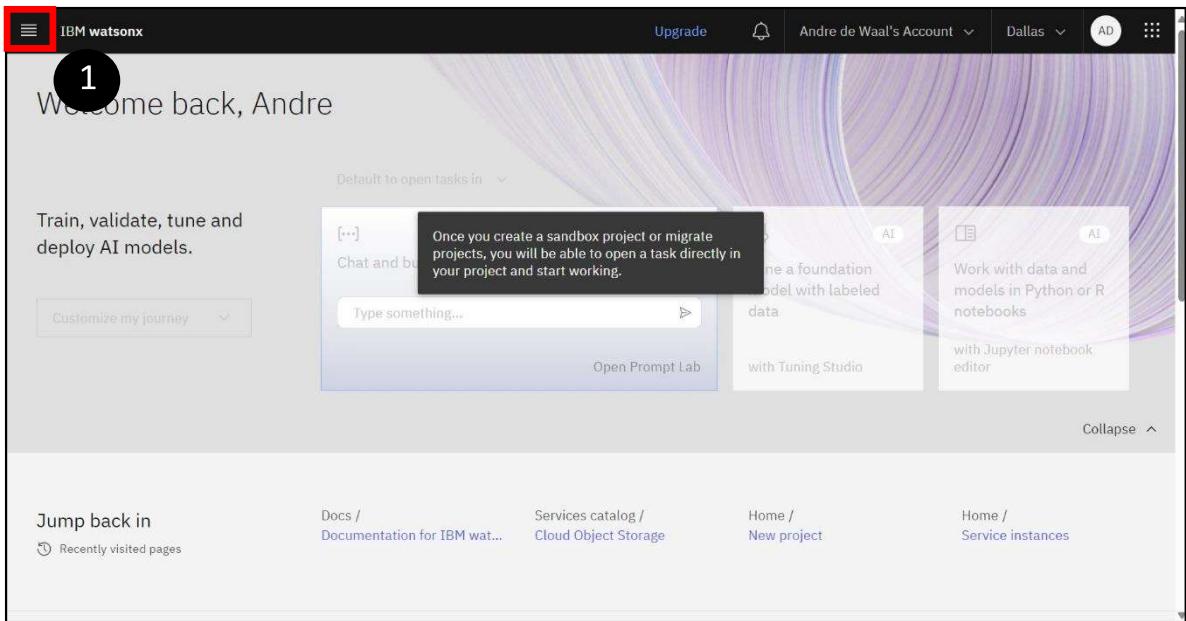
You successfully logged in to the WatsonX platform.

### 1.3 Setting Up the Lab Environment

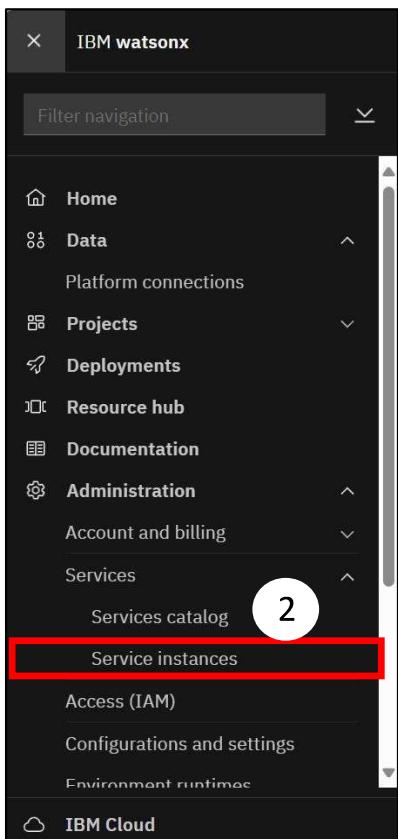
Before experimentation with the Prompt Lab can begin, a Watson Studio project needs to be created and a Watson Machine Learning instance needs to be associated to the project.

In this section, you create a Watson Studio project and associate a Watson Machine Learning service instance to a project.

1. Click the **Navigation Menu** (the four horizontal bars also known as the hamburger menu) in the upper-left quadrant of the screen.



2. Expand the **Services** twistie (also called the expander arrow) and select **Services instances** from the options.

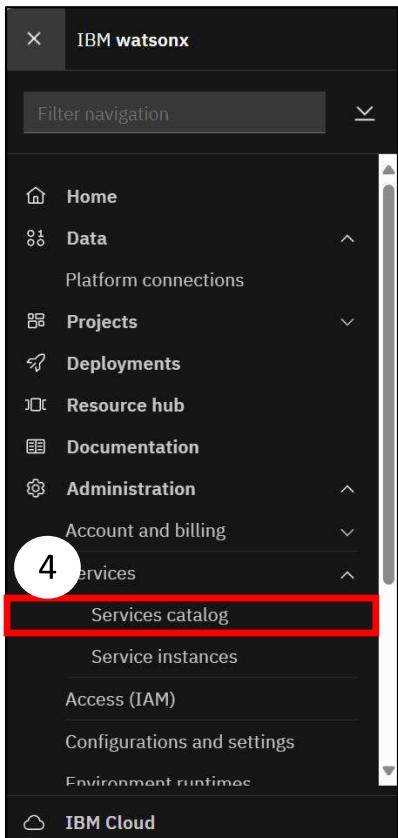


3. Review the service instances. If you have a **Watson Machine Learning** service instance running, click the **Main menu** icon to get back to the welcome screen and continue with

Step 8. Otherwise, continue with the next step to create a Watson Machine Learning instance.

Name	Group	Location	Product	Plan	Status	⋮
CloudObjectStorage	Default	Global	Cloud Object Storage	Lite(deprecated)	Active	⋮
WatsonMachineLearning	Default	Dallas	Watson Machine Learning	Lite	Active	⋮
WatsonStudio	Default	Dallas	Watson Studio	Lite	Active	⋮
watsonx.governance	Default	Dallas	watsonx.governance	Lite	Active	⋮

4. Expand the **Services** expander arrow and select **Services catalog** from the options.



5. Select the **Watson Machine Learning** tile under the **AI / Machine Learning** section.

Services catalog

AI / Machine Learning

Category

- AI / Machine Learning
- Analytics
- Databases
- Developer tools
- Integration
- Storage

**watsonx.governance**  
AI / Machine Learning • Analytics  
Accelerate responsibility, transparency and explainability in your data and AI workflows  
Lite • Free

**Watson Studio**  
AI / Machine Learning  
Develop sophisticated machine learning models using Notebooks and Jupyter tools to infuse AI through...

**Watson Machine Learning**  
AI / Machine Learning  
Deploy, manage and integrate machine learning models across your organization

- Verify that the **Lite** Plan is selected and click the blue **Create** button to create a Watson Machine Learning instance. If you already have an instance of Watson Machine Learning running, you can ignore this step.

Watson Machine Learning

Author: IBM SPSS • Date of last update: Apr 19, 2024 • Docs • API Docs

Create      About

Select a region

Select a region

Dallas

Pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or region: United States

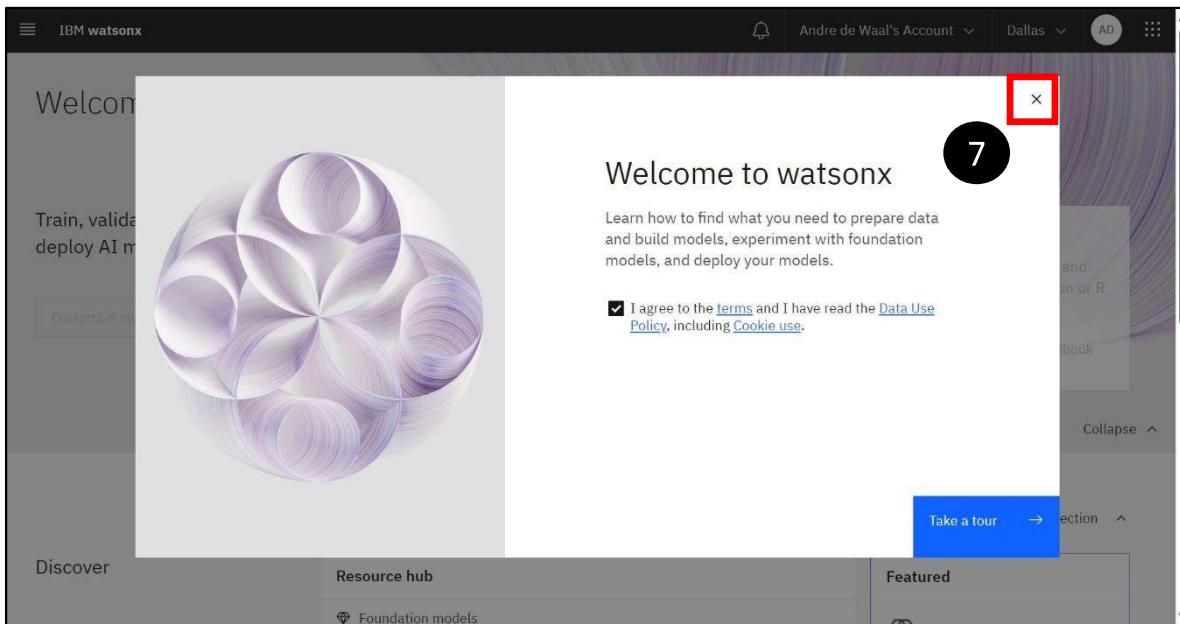
Plan	Features	Pricing
Lite	<b>Service instance</b> Instance includes: • 20 capacity unit-hours (CUH) per month • 50,000 tokens per month ----- <small>Foundation models (in Dallas, Frankfurt, and Tokyo regions only)</small>	Free

**Create**

View terms

Cancel

- You might be greeted by the **Welcome to watsonx** pop up window. Click **x** to close the window.



8. Scroll down to the **Recent work** section of the page. If you already have a Sandbox project created, click on the sandbox project to use it instead of creating a new project and continue with Step 15. Otherwise continue with the next step.

Welcome back, David

Open in: David's sandbox

Train, deploy, validate, and govern AI models responsibly.

Customize my journey

Recent work

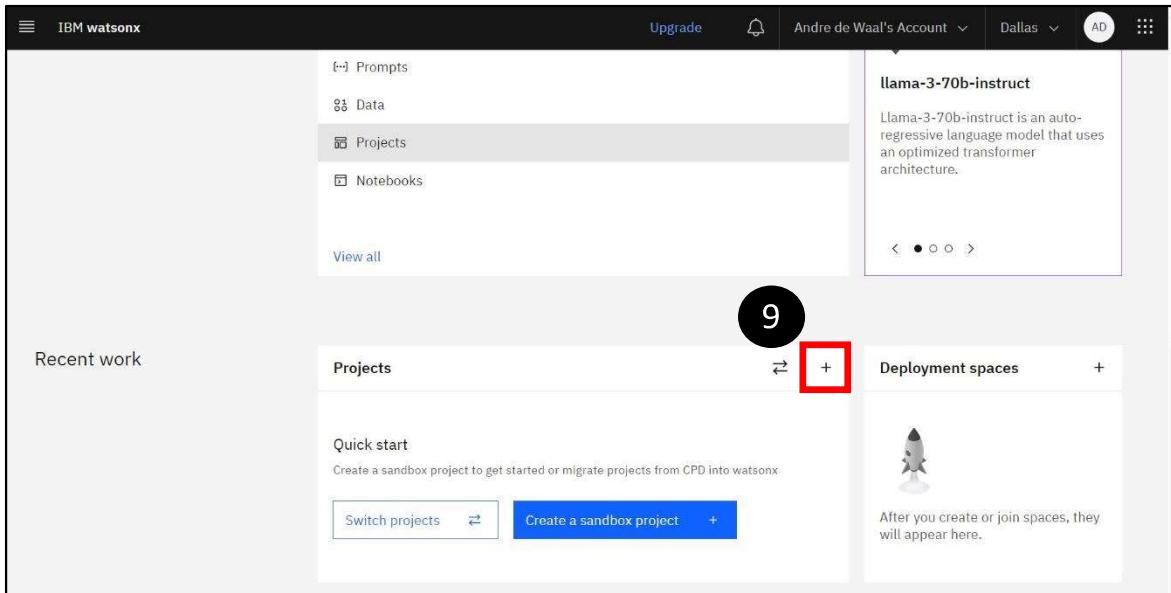
Projects

David's sandbox 1 h ago

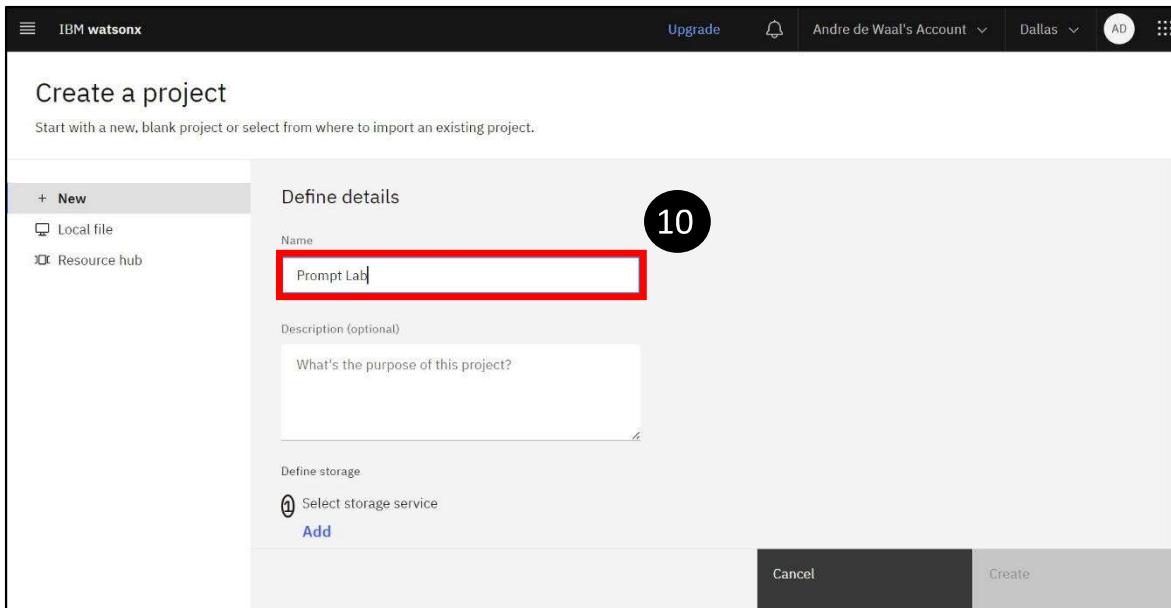
Recent spaces

After you create or join spaces, they will appear here.

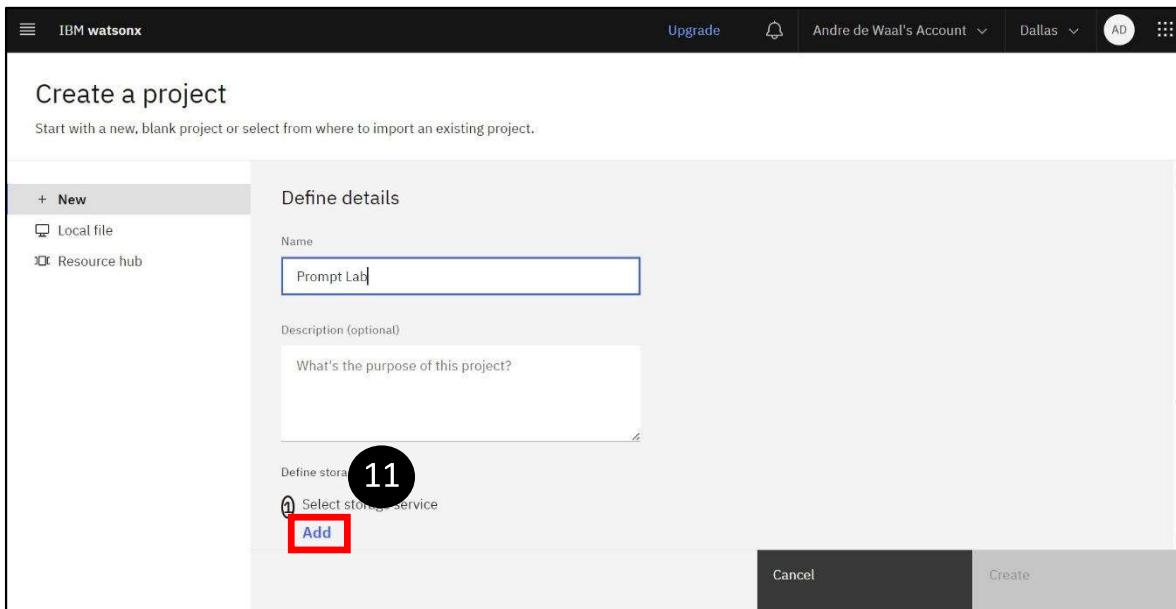
9. Scroll down in the **Recent work** section of the page and click **Create a new project** (+ in the upper-right of the projects area).



10. Enter **Prompt Lab** in the Name field.



11. Click **Add** to select a storage service. If you already have Cloud Object Storage configured (review the output of Step 3), it might be pre-selected, or you need to select from a list of available storage and continue with Step 14.



12. Scroll down and select the **Standard** Plan.

Plan	Features	Pricing
Standard	<p>Standard plan is our most popular Pay-as-You-Go option with no minimum fee, ideal for most enterprise workloads. It includes a Free Tier with 5GB of free storage for 12 months. To access the Free Tier, choose Smart Tier for your bucket storage class. The Free Tier has no cost; you pay only if your usage is beyond the free tier allowance.</p> <p>Free Tier allowance:</p> <ul style="list-style-type: none"><li>Storage up to 5GB/month</li><li>Up to 2000 Class A (PUT, COPY, POST, and LIST) requests/month</li><li>Up to 20,000 Class B (GET and all others) requests/month</li><li>Up to 10GB/month of data retrieval</li><li>Up to 5GB/month of Public outbound bandwidth</li><li>Applies to aggregate total across all smart tier buckets and Cloud Object Storage instances within your account.</li></ul> <p>The Standard plan is best suited for workloads that have large amount of storage and relatively small Outbound bandwidth. The plan offers flexible choices for storage class based on data access patterns (lower the cost, the less frequently data is accessed). The Standard plan bills for storage capacity, Outbound bandwidth, operational requests, and data retrieval, where applicable.</p>	<a href="#">View storage class pricing</a> <input checked="" type="checkbox"/>
One Rate	<p>One Rate plan offers a flat monthly charge that includes capacity, and built-in allowances for outbound bandwidth and data access. It is best suited for active workloads with large amounts of outbound bandwidth as a percent of their storage capacity.</p>	<a href="#">View storage class pricing</a>

A small black circle with a tick mark will appear in the upper-right corner of the selected

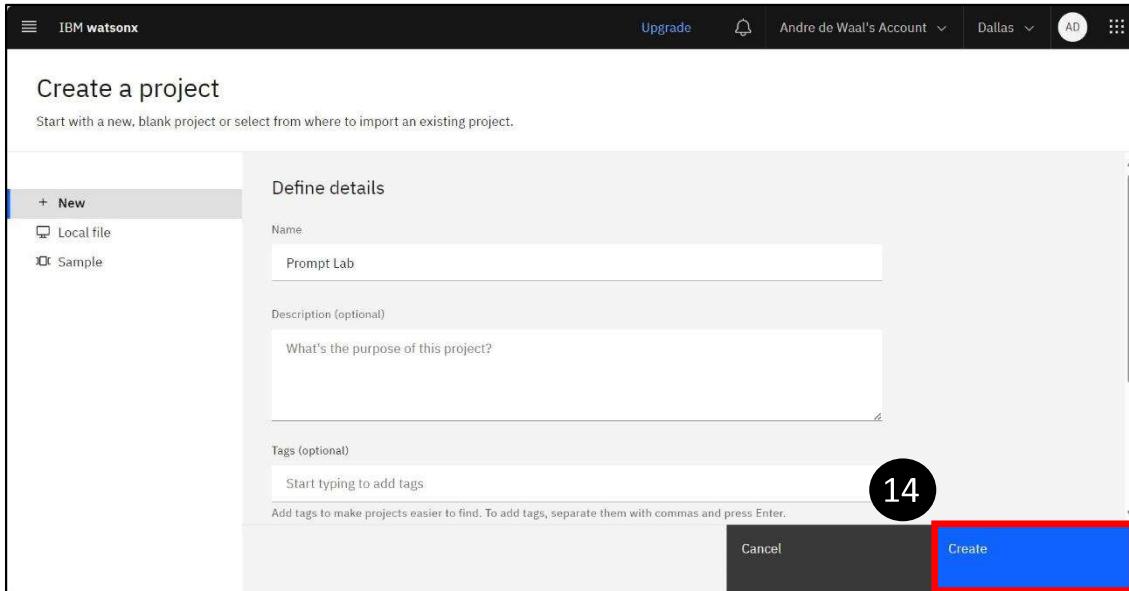
plan.

**Note:** From 1 July 2024 the Lite Plan for provisioning Cloud Object Storage is not available anymore. Select the Standard plan. If you do not want to pay for COS, stop and watch the recording in place of doing the lab!

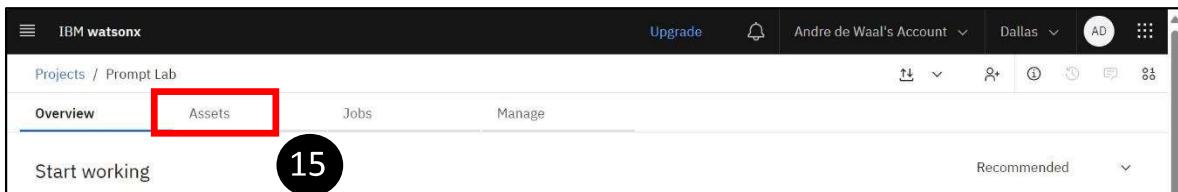
13. Click the **Create** button to create object storage.

The screenshot shows the IBM Cloud interface for creating Cloud Object Storage. On the left, a table compares two plans: 'Standard' and 'One Rate'. The 'Standard' plan is selected, indicated by a checked checkbox in the 'Pricing' column. The 'Features' column lists various allowances, including Free Tier up to 5GB/month, 2000 Class A requests/month, 20,000 Class B requests/month, 10GB/month of data retrieval, and 5GB/month of Public outbound bandwidth. The 'Description' section notes that the Standard plan is best suited for workloads with large storage and small outbound bandwidth. The 'One Rate' plan offers a flat monthly charge for capacity and bandwidth. On the right, the 'Summary' panel shows the service type as 'Cloud Object Storage', region as 'Global', plan as 'Standard', service name as 'Cloud Object Storage-tt', and resource group as 'Default'. At the bottom right, a large blue button labeled 'Create' is highlighted with a red box. A circular badge in the top right corner contains the number '14'.

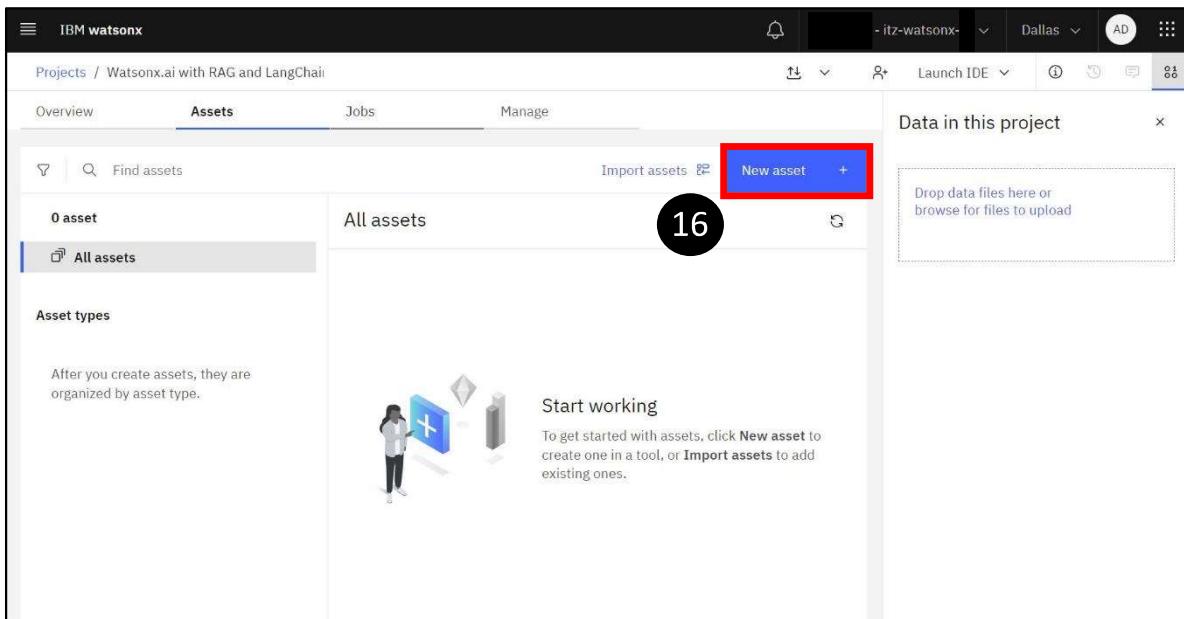
14. Click the **Create** button to create the Prompt Lab project and to attach the object storage to the project.



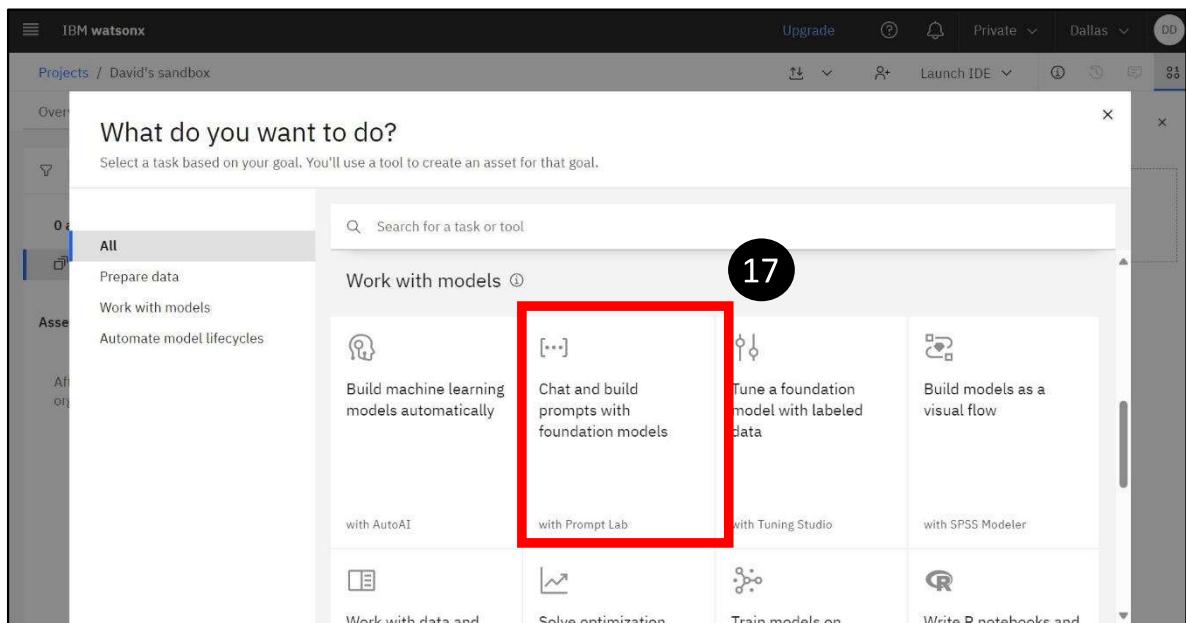
15. Select the **Assets** tab.



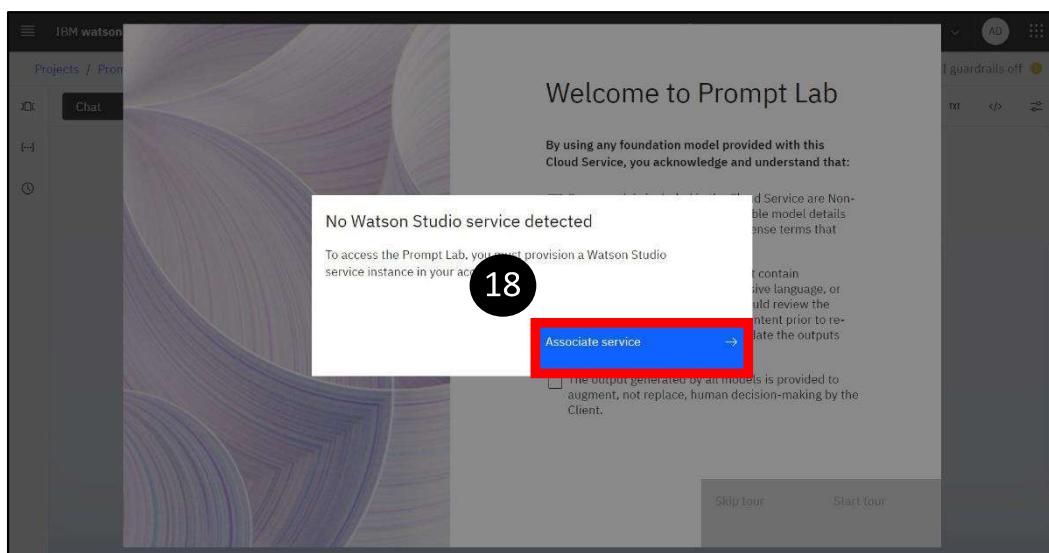
16. Note that the project currently has no assets. Click the blue **New asset** button.



17. Scroll down and select the Chat and build prompts with foundation models tile.

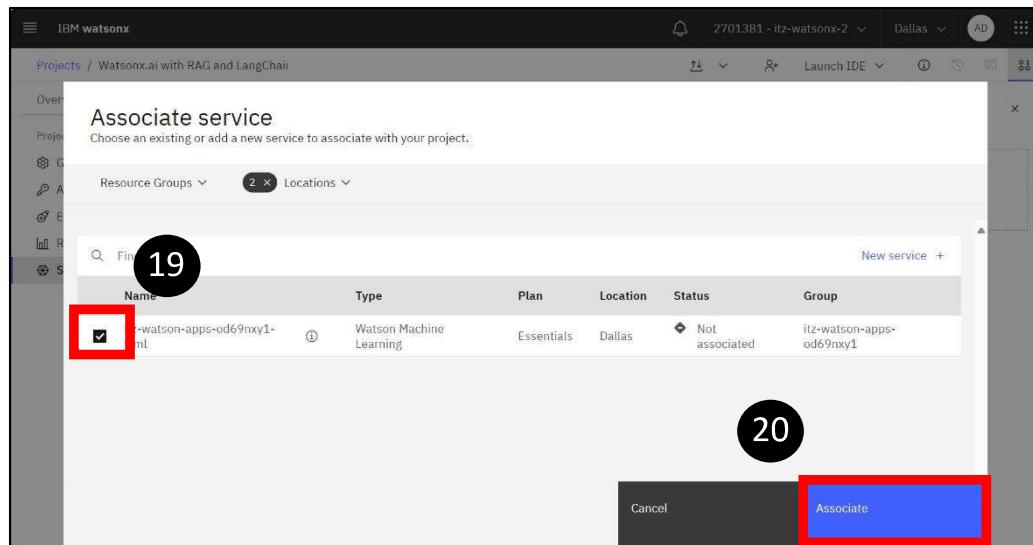


18. Depending on whether you created a Sandbox project or not, you may get the following message. If you created a Sandbox project, you have a Watson Machine Learning service associated with the project and you continue with Step 21. If no machine learning service instance is currently associated to this project, you get the following message. Although the Watson Machine Learning service exists, the service instance needs to be associated to your project to be able to experiment with the Prompt Lab. Click the blue **Associate service** button.

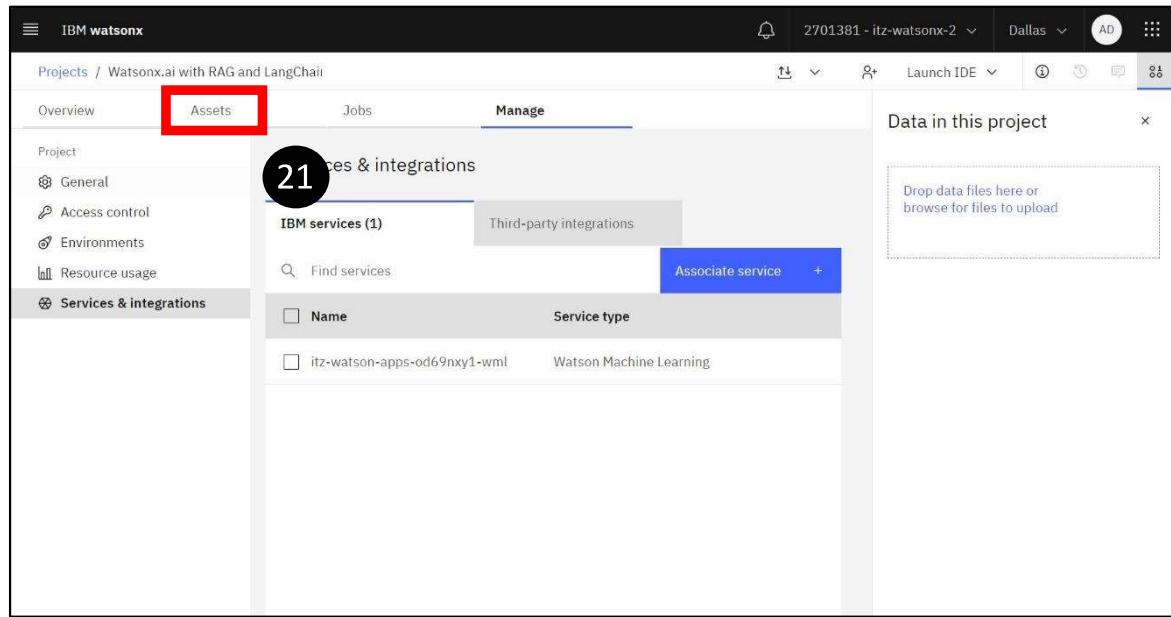


19. Select the checkbox next to your **Watson Machine Learning** service instance.

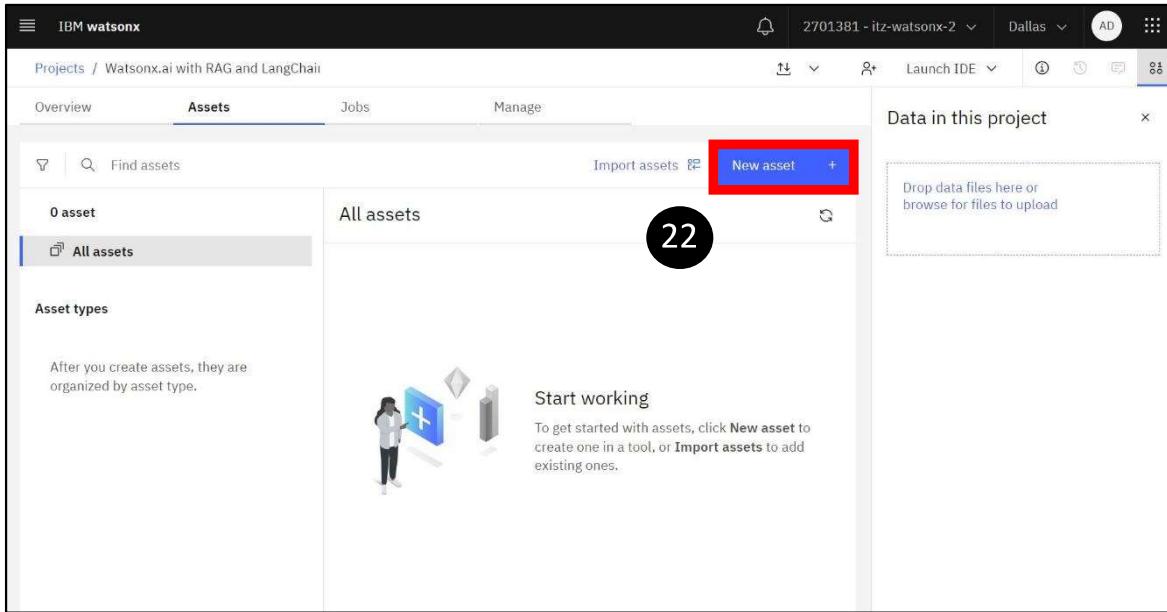
20. Click the blue **Associate** button.



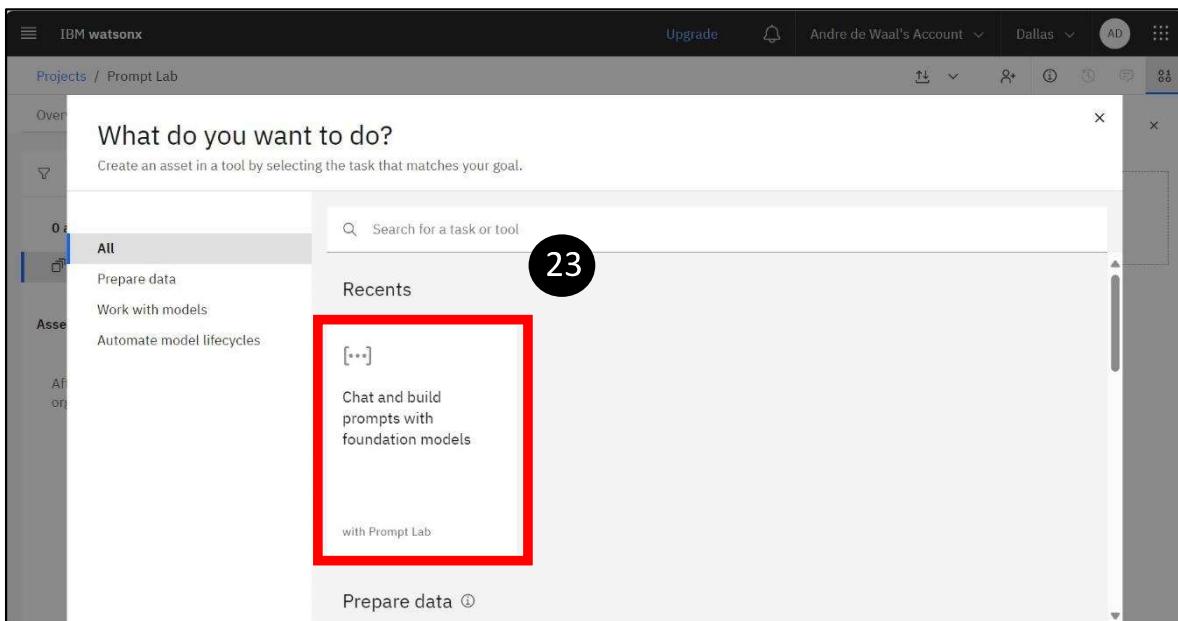
21. Select the **Assets** tab to get back to the project assets screen.



22. The assets screen displays. Notice that you still have no assets associated with this project. Click the blue **New asset** button.

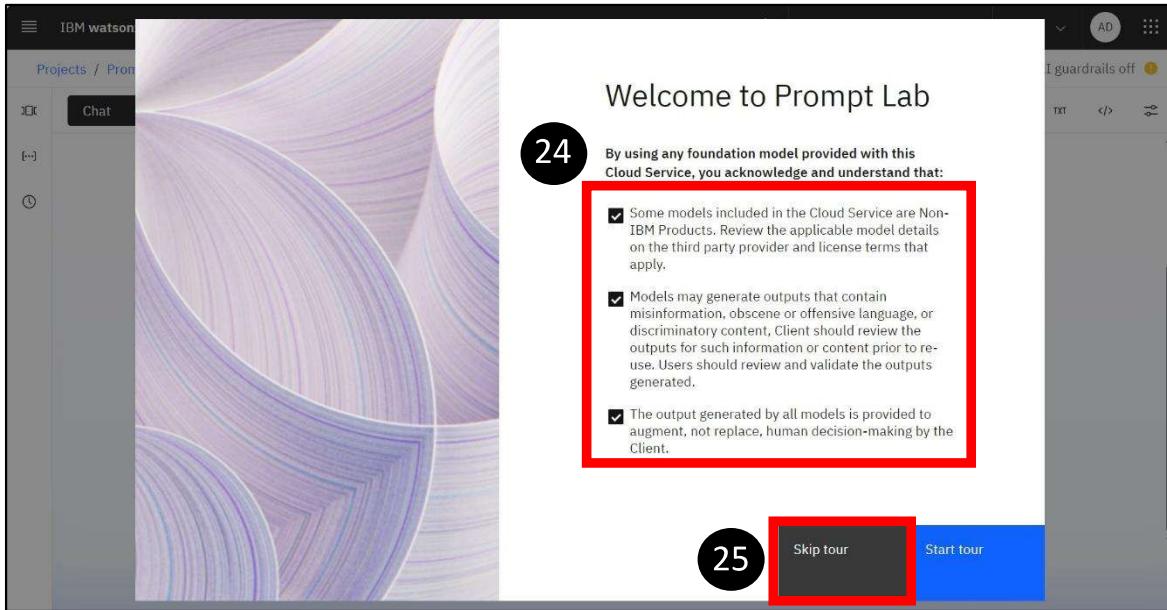


23. Scroll down and select the **Chat and build prompts with foundation models** tile.



24. Select the 3 **check boxes** to accept the conditions to use this service.

25. Click the black **Skip tour** button.



The Prompt Lab opens.

IBM watsonx

Projects / Prompt Lab / Prompt Lab

Chat Structured Freeform

watsonx 12:49 PM

Customize your chat

Before you start chatting, you can change the model, edit the system prompt, and adjust model parameters.

Quick start samples

Type something...

The Prompt Lab, the Tuning Studio, and Jupyter Notebooks with Python code are now available for solving several use cases.

## 2. Prompt engineering

### Estimated time

30:00 minutes (to complete this section)

### Overview

This exercise covers the following tasks:

- Experimenting with the prompt Lab
- Creating a series of ever more complex prompts
- Inspecting the Python code that the Prompt Lab generates
- Changing the default Prompt Lab settings

### Objectives

This exercise introduces you to the Prompt Lab in watsonx. First, you experiment with simple prompts to get a feel for the environment. Second, you create more elaborate prompts to steer the underlying foundation model in the right direction. Third, you inspect the underlying Python code generated by the lab. Fourth, you change the underlying foundation model and experiment with different lab settings.

After this exercise, you should be able to set up and experiment with prompts in the Prompt Lab.

### Requirements

- Access to IBM Cloud Pak for Data, IBM watsonx.
- Knowledge of working with graphical user interfaces.
- Working knowledge of Python and Jupyter Notebooks.

### 2.1 Writing your first prompt

Verify that you are still logged in to the IBM Cloud and watsonx. If not, go back to Exercise 1 and revisit how to log in to the IBM Cloud and watsonx.

In this exercise you assume the role of Erik, the AI Specialist, and you follow along as he explores watsonx.ai. Erik is new to prompt engineering, so he first explores the watsonx.ai interface with some basic prompts and gradually increases the complexity of the prompts as he gains more experience.

After the exercise, you will be able to recommend some use cases for generative AI and foundation models to the company.

## Section 1. Zero-shot prompting

In this section, you are an AI specialist who creates a couple of basic prompts. Creating basic prompts without any examples or any other additional information is also referred to as zero-shot prompting.

You can create prompts in a structured or a freeform way. Using the structured approach, you give the model instructions on what to do, examples of how to format the response, and text that you want the model to process. All this information is then concatenated to form a prompt that is passed to the selected foundation model. In the freeform approach, your text is formatted in one text box.

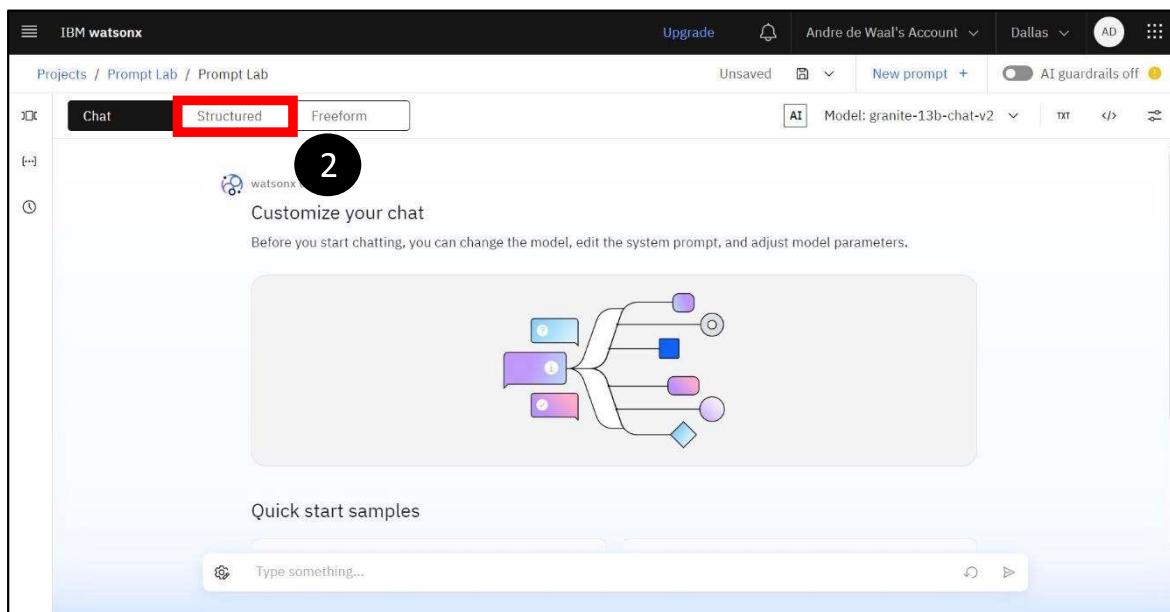
Since you new to prompt engineering, you starts with the structured approach and interface (which is the default option when you open the Prompt Lab).

1. Click **AI Guardrails** in the upper-right quadrant to set **AI guardrails** on.

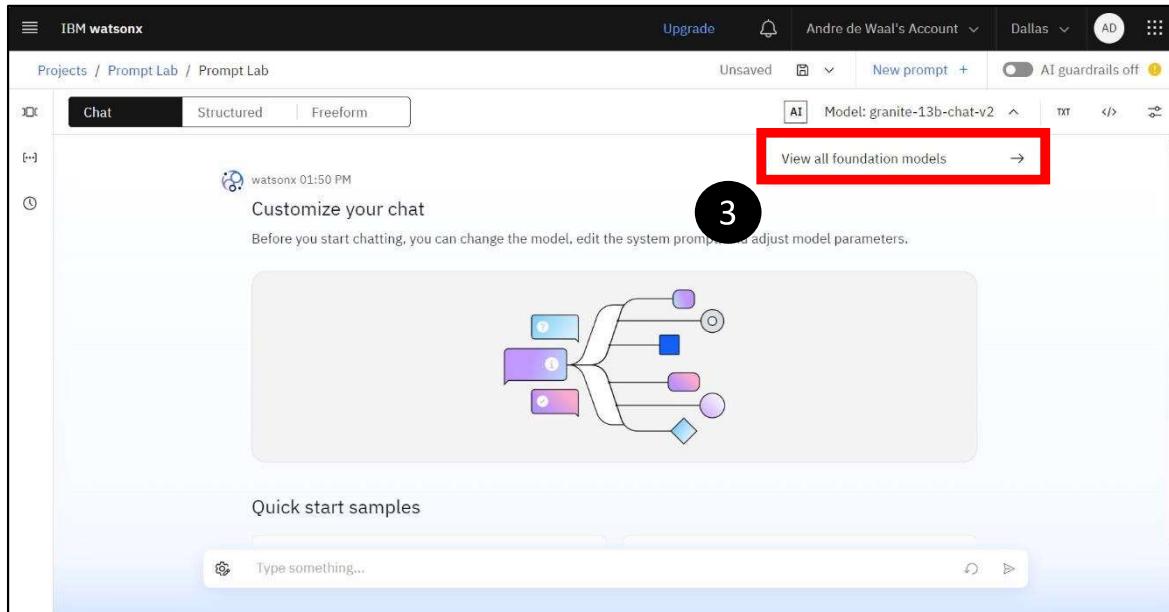


This setting removes any potentially harmful input and output text, such as hate speech, abuse, or profanity (HAP). The default setting is for the guardrails to be off (no filtering of input or output text).

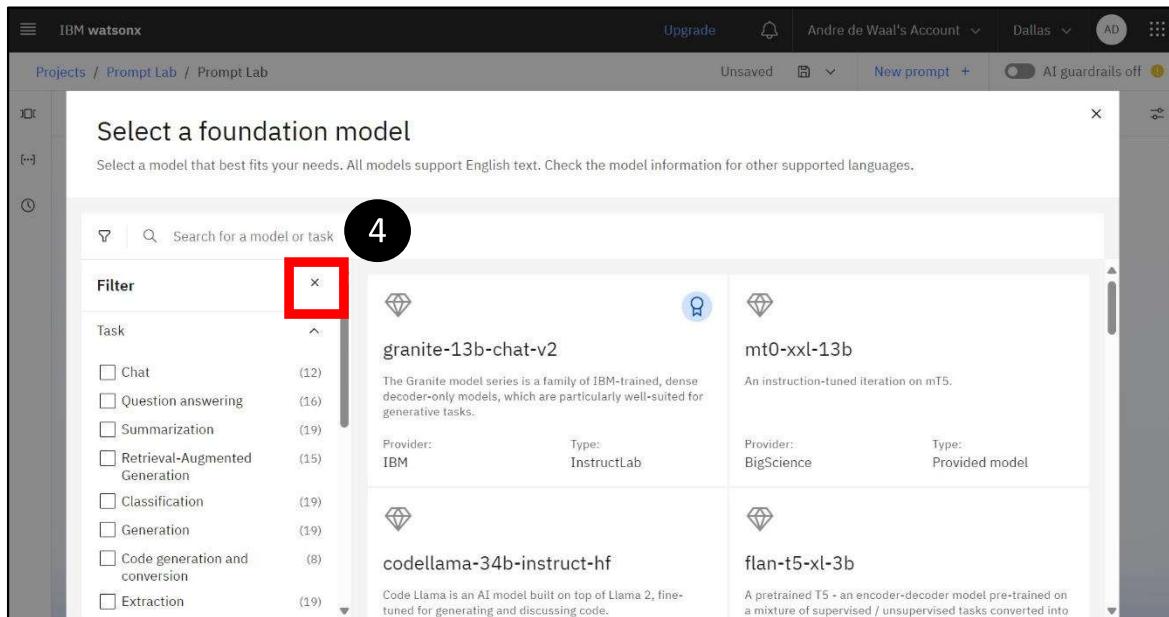
2. Select the **Structured** interface.



3. Expand the model menu and select **View all foundation models**.



4. If you see the Filter panel, close the panel by clicking the x next to Filter.



5. Scroll down and select the **flan-ul2-20b** model tile from the available foundation models.

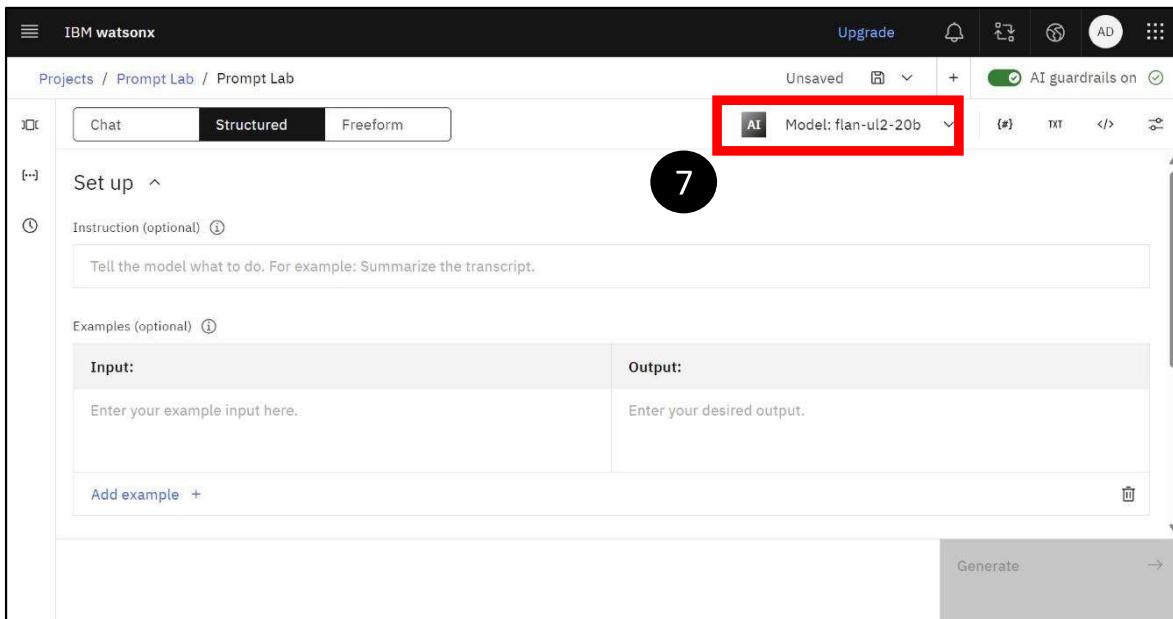
The screenshot shows the 'Select a foundation model' page in the IBM Watsonx interface. At the top, there are navigation links for 'Projects / Watsonx.ai with RAG and LangCh...', 'Prompt Lab', and a status bar with 'Upgrade', 'IBM', 'Dallas', and 'AD'. Below the header, there's a search bar with placeholder text 'Search for a model or task'. The main area displays a grid of foundation models. The first model, 'flan-ul2-20b', is highlighted with a red box and a large number '5' above it. Other models listed include 'merlinite-7b', 'mixtral-8x7b-instruct-v01-q', 'granite-13b-instruct-v2', 'granite-20b-multilingual', and 'granite-7b-lab'. Each model entry includes its name, provider, type, and a brief description.

Provider:	Type:
Code Llama	Provided model
Provider:	Type:
Google	Provided model
Provider:	Type:
Mistral AI, tuned by IBM	InstructLab
Provider:	Type:
Mistral AI, tuned by IBM	Provided model
Provider:	Type:
Granite	Provided model
Provider:	Type:
Granite	Provided model
Provider:	Type:
Granite	Provided model

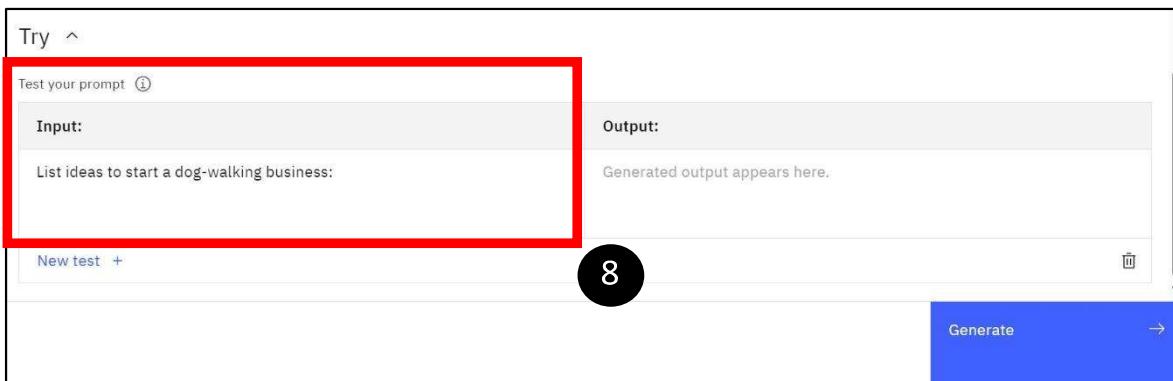
6. Inspect the flan-ul2-20b model card and click the blue **Select model** button.

The screenshot shows the IBM WatsonX Prompt Lab interface. At the top, there's a navigation bar with 'IBM watsonx' on the left, followed by 'Upgrade', a bell icon, 'IBM', 'Dallas', and 'AD'. Below the navigation is a breadcrumb path: 'Projects / Watsonx.ai with RAG and LangCh... / Prompt Lab'. To the right of the path are 'Unsaved', a file icon, a dropdown arrow, 'New prompt', a plus sign, and a green button labeled 'At guardrails on' with a gear icon. The main content area has a title 'flan-ul2-20b' with a close button 'x'. Below the title is the text 'Provider: Google | Type: Provided model'. A horizontal row of buttons includes 'Question answering', 'Summarization', 'Retrieval-Augmented Generation', 'Classification', 'Generation', and 'Extraction'. A note below these buttons states: 'Note: This model is a Non-IBM Product governed by a third-party license that may impose use restrictions and other obligations. By using this model you agree to these terms.' with a 'Read terms' link and a close button 'x'. The central part of the screen is titled 'Model card for Flan-UL2'. At the bottom, there's a footer with a back button, a large red box containing a 'Select model' button, and a number '6' in a black circle.

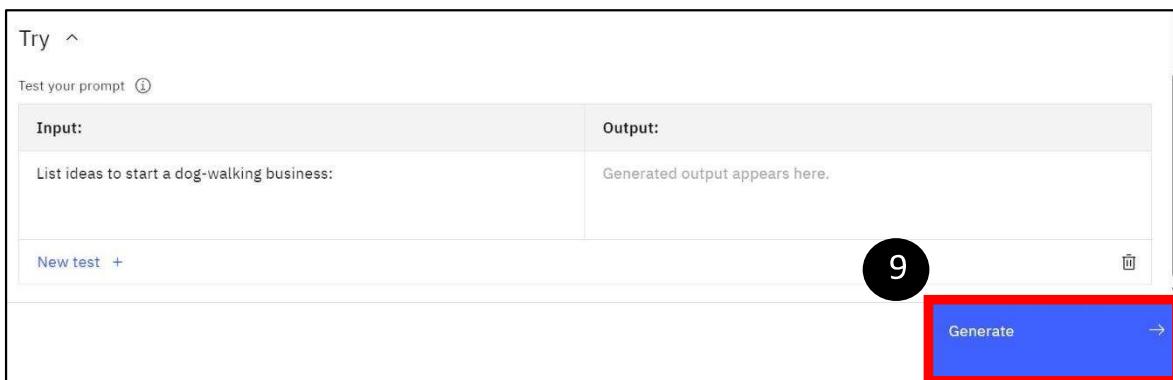
- Your Prompt Lab is updated. **Verify** that the **flan-ul2-20b** foundation model is the selected model for this Prompt session.



8. Scroll to the bottom of the displayed Prompt Lab page and type in the following prompt in the **Input:** text box under the **Try** section: **List ideas to start a dog-walking business:**



9. Click the blue **Generate** button.



The output of the foundation model is given in the **Output:** text box. Inspect the output.

The screenshot shows the IBM WatsonX Prompt Builder interface. At the top, there's a "Try ^" button and a "Test your prompt" field with a help icon. Below that is a table with two columns: "Input:" and "Output:". The "Input:" column contains the text "List ideas to start a dog-walking business:". The "Output:" column contains the generated response: "Start a dog-walking business by identifying a need in your community." An "AI" icon is in the top right corner of the output box. At the bottom of the interface, there's a status bar with "Stop reason: End of sequence token encountered", "Tokens: 19 input + 19 generated = 38 out of 4096", "Time: 1.2 seconds", a "Clear output" button, a "Generate" button, and a blue "→" button.

The response is underwhelming. After exploring the sample prompts from the IBM Documentation [Tips for writing foundation model prompts: prompt engineering — Docs | IBM watsonx](#), You decide to take “baby steps” and gradually increase the complexity of the prompt, incorporating what he learned from studying the tips for writing prompts.

---

**Information:** Each time you enter a prompt, your “input tokens” and “generated tokens” will update in the lower-left quadrant of the interface. Tokens are an important concept to understand as they constrain the performance of your model and determine the cost of using foundation models. Tokens are not a 1:1 match with words in natural language. On average, one token is equal to 4 characters. Before sending your prompt to a model, the prompt's text is tokenized or broken into smaller subsets of characters than can be better understood by the model. The correlation between words and tokens is complex. A single word might be broken into multiple tokens, depending on context (such as where the word appears in a sentence, or what the surrounding words are). Spaces, newline characters, and punctuation might be included in tokens. The way words are broken into tokens varies from language to language and from model to model.

**Important:** It is important to monitor your token usage to know how much information you are feeding into the model with each prompt, and how much text is generated for you. Depending on the model selected in Prompt Builder, you see a max of 2048 or 4096 tokens. Keep in mind that the more expressive you are with your prompt instructions, the less room the model has to respond back to you. The number of tokens used (input plus generated text) can be found in the lower-left quadrant of the interface.

Stop reason: End of sequence token encountered  
Tokens: 159 input + 60 generated = 219 out of 4096 | Seed: 216726205  
Time: 2.7 seconds

10. A good way to receive a structured response is to include a cue to start the response in the input structure. For example, add the two characters, "1.", to the prompt on a new line. This small change might improve the response. Click **New test +** underneath your prompt.

Try ^

Test your prompt ⓘ

Input:	Output:
List ideas to start a dog-walking business: 10	Start a dog-walking business by identifying a need in your community.

New test +

Stop reason: End of sequence token encountered.  
Tokens: 19 input + 19 generated = 38 out of 4096  
Time: 1.2 seconds

Clear output ↻ Generate →



11. Type in the following prompt in the **Input:** text box under the **Try** section:

**List ideas to start a dog-walking business:**

1.

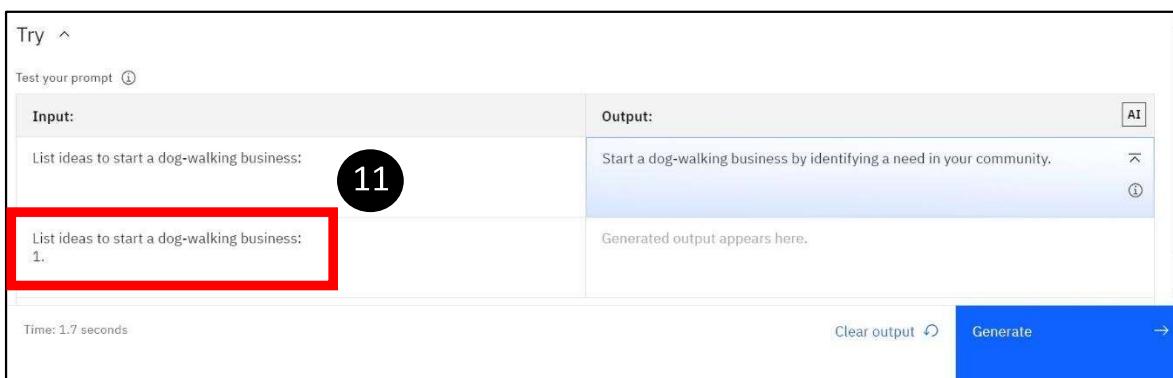
Try ^

Test your prompt ⓘ

Input:	Output:
List ideas to start a dog-walking business: 11	Start a dog-walking business by identifying a need in your community.
List ideas to start a dog-walking business: 1.	Generated output appears here.

Time: 1.7 seconds

Clear output ↻ Generate →



12. Click the blue **Generate** button.

Try ^

Test your prompt ⓘ

Input:	Output:
List ideas to start a dog-walking business:	Start a dog-walking business by identifying a need in your community.
List ideas to start a dog-walking business: 1.	Generated output appears here.

Time: 1.7 seconds

Clear output ↻ Generate →



The response is still underwhelming. You decide to investigate the current model parameters and their effect on the response.

The screenshot shows the AI interface with two test examples. The first example, "List ideas to start a dog-walking business:", has an output of "Start a dog-walking business by identifying a need in your community." The second example, "List ideas to start a dog-walking business:  
1.", has an output of "Start a dog-walking business." A red box highlights the "AI" icon in the top right corner of the output section. At the bottom, there is a "Generate" button.

13. Click the **Model parameters** icon in the upper-right quadrant (two horizontal lines with offset circles).

The screenshot shows the AI interface with the same two test examples. A large black circle with the number "13" is drawn around the "Model parameters" icon in the top right corner of the interface. The interface includes tabs for Chat, Structured, and Freeform, and a status bar at the bottom indicating "Time: 1.2 seconds".

14. Scroll down and inspect the Stopping criteria section.

The screenshot shows the AI interface with the 'Structured' tab selected. In the 'Output' section, there is a single item: 'Start a dog-walking business by identifying a need in your community.' A large black circle labeled '14' is positioned over the 'Model parameters' panel on the right. This panel includes sections for 'Decoding' (set to 'Greedy'), 'Repetition penalty' (set to 1), and 'Stopping criteria'. The 'Stopping criteria' section is highlighted with a red box and contains fields for 'Stop sequences' (empty), 'Min tokens' (0), and 'Max tokens' (200). At the bottom of the interface, there is a blue 'Generate' button.

The reason for the short result might be the minimum and the maximum number of tokens settings. The **Min tokens** is set to **0**, which does not force the model to generate a lengthy response (on average 1 token is equivalent to 4 characters).

15. Delete the previous prompts and add a new test with the same prompt as the one you just used (with the **1.** added). Change the **Min tokens** value to **50** and the **Max tokens** to **100** to “force” the foundation model to generate a longer response (more characters).

The screenshot shows the AI interface with the 'Structured' tab selected. In the 'Output' section, there is a message: 'Generated output appears here.' A large black circle labeled '15' is positioned over the 'Model parameters' panel on the right. This panel includes sections for 'Repetition penalty' (set to 1) and 'Stopping criteria'. The 'Stopping criteria' section is highlighted with a red box and contains fields for 'Stop sequences' (empty), 'Min tokens' (50), and 'Max tokens' (100). At the bottom of the interface, there is a blue 'Generate' button.

16. Click the blue **Generate** button.

Try ^

Test your prompt ⓘ

Input:	Output: AI
List ideas to start a dog-walking business: 1.	Generated output appears here.

New test +

Time: 1.2 seconds

Clear output ↻ Generate →

Repetition penalty: 1 ● 2 1

Stopping criteria ⓘ

Stop sequences: +

Min tokens: 50 Max tokens: 100

Enter up to 6 sequences to stop output

The results improved somewhat, but there is much repetition of the output **Start a dog-...** **business,** in the response.

Try ^

Test your prompt ⓘ

Input:	Output: AI
List ideas to start a dog-walking business: 1.	Start a dog-walking business. 2. Start a dog-sitting business. 3. Start a dog-boarding business. 4. Start a dog-training business. 5. Start a dog-grooming business. 6. Start a dog-sitting business. 7. Start a dog-walking business. 8. Start a dog-boarding business. 9. Start a dog-training business. 10. Start a dog-sitting business.

New test +

Stop reason: End of sequence token encountered  
Tokens: 20 input + 100 generated = 120 out of 4096  
Time: 3.9 seconds

Clear output ↻ Generate →

Greedy Sampling ⓘ

Repetition penalty: 1 ● 2 1

Stopping criteria ⓘ

Stop sequences: +

Min tokens: 50 Max tokens: 200

Enter up to 6 sequences to stop output

17. After further consideration, you decide to increase the Repetition penalty from **1** to **2**. Changing the Repetition penalty in the Model parameters panel from **1** to **2** attempts to minimize repetition. Slide the **Repetition penalty slider** from **1** to **2**, or directly type in **2** in the Repetition penalty numeric box (beside the slider).

Try ^

Test your prompt ⓘ

Input:	Output: AI
List ideas to start a dog-walking business: 1.	Start a dog-walking business. 2. Start a dog-sitting business. 3. Start a dog-boarding business. 4. Start a dog-training business. 5. Start a dog-grooming business. 6. Start a dog-sitting business. 7. Start a dog-walking business. 8. Start a dog-boarding business. 9. Start a dog-training business. 10. Start a dog-sitting business.

Stop reason: End of sequence token encountered  
Tokens: 138 input + 100 generated = 238 out of 4096  
Time: 6.2 seconds

Clear output ↻ Generate →

Repetition penalty: 1 ● 2 2

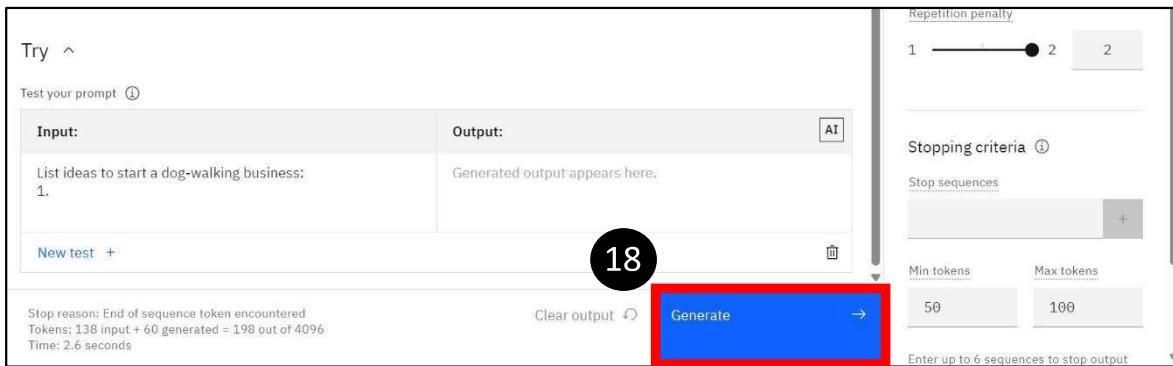
Stopping criteria ⓘ

Stop sequences: +

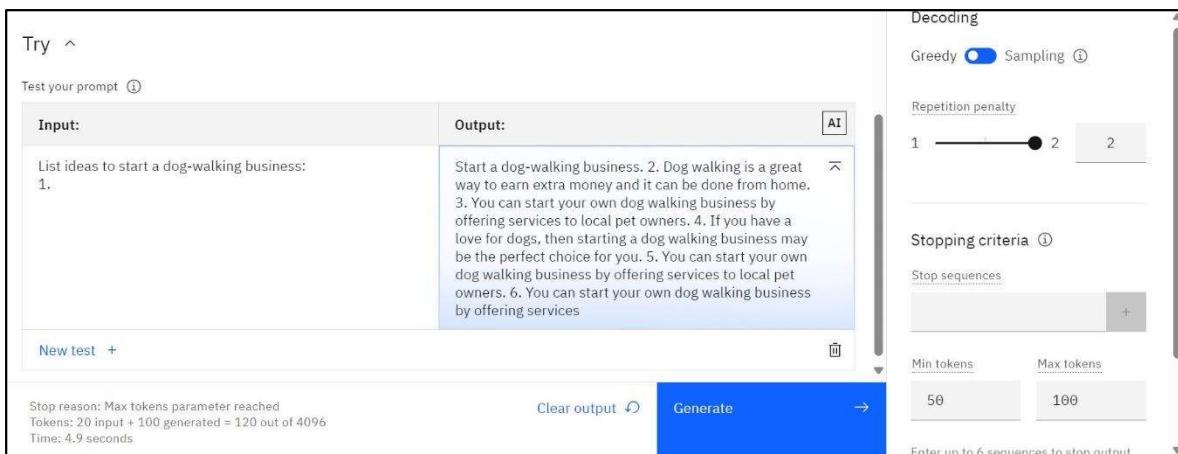
Min tokens: 50 Max tokens: 200

Enter up to 6 sequences to stop output

18. Clear the output (next to the Generate button) and click the blue **Generate** button.



As you can see, increasing the repetition penalty parameter creates a dramatic improvement over the previous responses.



## Section 2: Single-shot and two-shot prompting

You also learned from reading the *Tips for writing foundation model prompts* that adding examples might improve the LLM response. To receive a higher-quality response, you should provide an example of the kind of response you want. This approach is called *single-shot* or *one-shot* prompting.

1. In the **Setup** section, Under **Examples**, add the expected Input and Output on starting a lemonade business:

**Input:** List ideas to start a lemonade business:

**Output:**

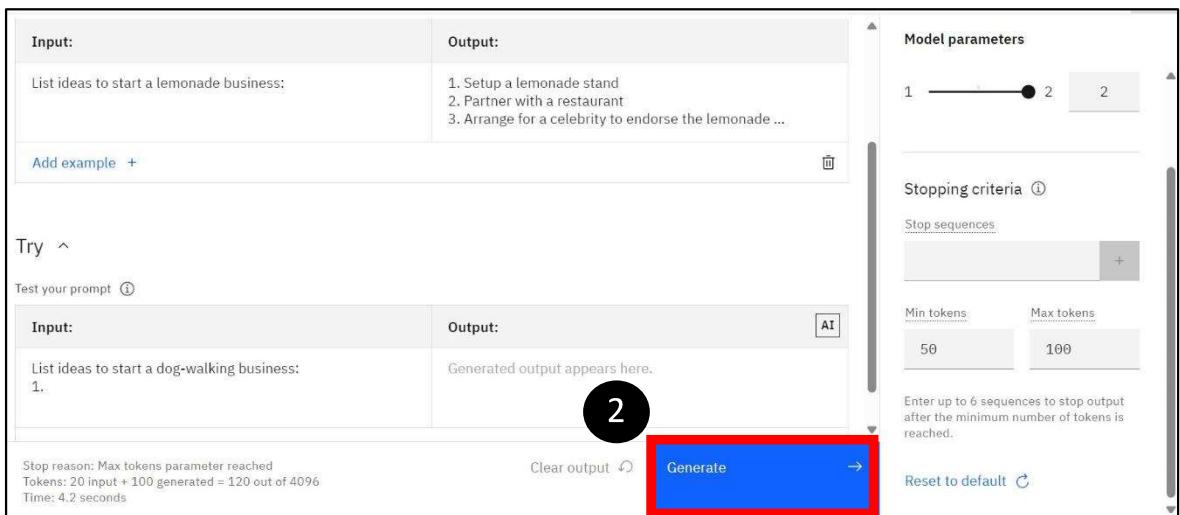
1. Setup a lemonade stand
2. Partner with a restaurant

### 3. Arrange for a celebrity to endorse the lemonade stand



The screenshot shows an AI model interface. In the 'Input' field, the text 'List ideas to start a lemonade business:' is entered. In the 'Output' field, the generated response is: '1. Setup a lemonade stand  
2. Partner with a restaurant  
3. Arrange for a celebrity to endorse the lemonade ...'. A red box highlights this output section. On the right side, there are 'Model parameters' including 'Sampling' (selected), 'Repetition penalty' (set to 2), and 'Stopping criteria' (with 'Min tokens' at 50 and 'Max tokens' at 100). The 'Generate' button is blue.

2. Clear the output (next to the Generate button) and click **Generate**.



The screenshot shows the same AI model interface. The 'Input' field now contains 'List ideas to start a dog-walking business:'. The 'Output' field displays 'Generated output appears here.' A red box highlights the 'Generate' button, which is now white. A black circle with the number '2' is placed over the 'Generate' button. The rest of the interface remains the same, including the 'Model parameters' on the right.

The response is more concise, but not necessarily an improvement over the previous response.

The screenshot shows a user interface for generating AI text. In the 'Input' field, the user has typed: "List ideas to start a dog-walking business:  
1.". In the 'Output' field, the AI has generated the following response: "Start by walking dogs in your neighborhood. 2. Offer dog-sitting services. 3. Offer to walk dogs for people who are away on vacation. 4. Offer to walk dogs for people who work long hours. 5. Offer to walk dogs for people who have surgery." To the right of the input and output fields are various configuration options: 'Sampling' (set to 'Greedy'), 'Repetition penalty' (set to 1.5), 'Stopping criteria', 'Stop sequences', 'Min tokens' (set to 50), and 'Max tokens' (set to 100). At the bottom, there are buttons for 'Clear output' and 'Generate'.

- Add another example to this section, this time on how to start a landscaping business. Click the **Add example +** button (below the **List ideas to start a dog-walking business:** example) to add space for a new example.

- Under **Examples**, add the expected **Input** and **Output** on starting a lemonade business:

**Input:** List ideas to start a landscaping business:

**Output:**

- Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows
- Buy some inexpensive equipment like a lawn mower, rakes, shovels, etc.
- Purchase used lawn furniture from estate sales or yard sales at very low cost

The screenshot shows the 'Examples (optional)' section of the AI tool. It contains two examples. The first example is for a landscaping business, with its input and output highlighted with a red box. The second example is for a lemonade business. Below the examples is a blue button labeled 'Add example +'.

- Clear the output (next to the Generate button) and click the blue **Generate** button.

The screenshot shows the Prompt Lab interface with two examples of prompts and their outputs. The first example is "List ideas to start a lemonade business:" followed by a list of three items. The second example is "List ideas to start a landscaping business:" followed by a list of two items. Below these examples is a "Try" section with a "Test your prompt" input field containing "List ideas to start a dog-walking business:". The output for this prompt is "Generated output appears here." To the right of the output is a "Model parameters" sidebar with options for Decoding (Greedy selected), Repetition penalty (set to 2), Stopping criteria (empty), and a "Generate" button highlighted with a red box and a large blue circle labeled "5".

The result is even better and contains more creative ideas. The response contains a lot of detail, and it is well structured.

This screenshot shows the same Prompt Lab interface after adding an instruction. The "Input" field now includes the text "Act as an entrepreneur starting a small one-person business. List ideas to start a successful business." The "Output" field displays a detailed list of five ideas for starting a dog-walking business, including walking dogs in neighborhoods, advertising online, walking dogs at parks, offering dog-sitting services, and taking dogs for rides. The "Model parameters" sidebar remains the same, with the "Generate" button highlighted by a red box and a large blue circle labeled "5".

6. Reviewing the Prompt Lab interface and thinking back to the tips for writing good prompts, you noticed that you did not provide any instructions in the Instruction optional box. An instruction is an imperative statement that tells the model what to do. In this case, the imperative statement refers to an unconditional action to be taken by the foundation model. Add the following instruction to the **Instruction (optional)** box:

**Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.**

The important keywords trying to steer the model in the right direction are “Act”, “entrepreneur”, “one-person” and “successful”.

Set up ^

Instruction (optional) ⓘ

Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.

6

7. Clear the output and click the blue **Generate** button.

Set up ^

Instruction (optional) ⓘ

Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.

Examples (optional) ⓘ

Input:	Output:
List ideas to start a lemonade business:	1. Setup a lemonade stand 2. Partner with a restaurant 3. Arrange for a celebrity to endorse the lemonade ...
List ideas to start a landscaping business:	1. Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows 2. Buy some inexpensive equipment like a lawn mow...

Add example +

Stop reason: End of sequence token encountered  
Tokens: 138 input + 60 generated = 198 out of 4096  
Time: 3.1 seconds

Model parameters

Decoding

Greedy  Sampling ⓘ

Repetition penalty

1  2

Stopping criteria ⓘ

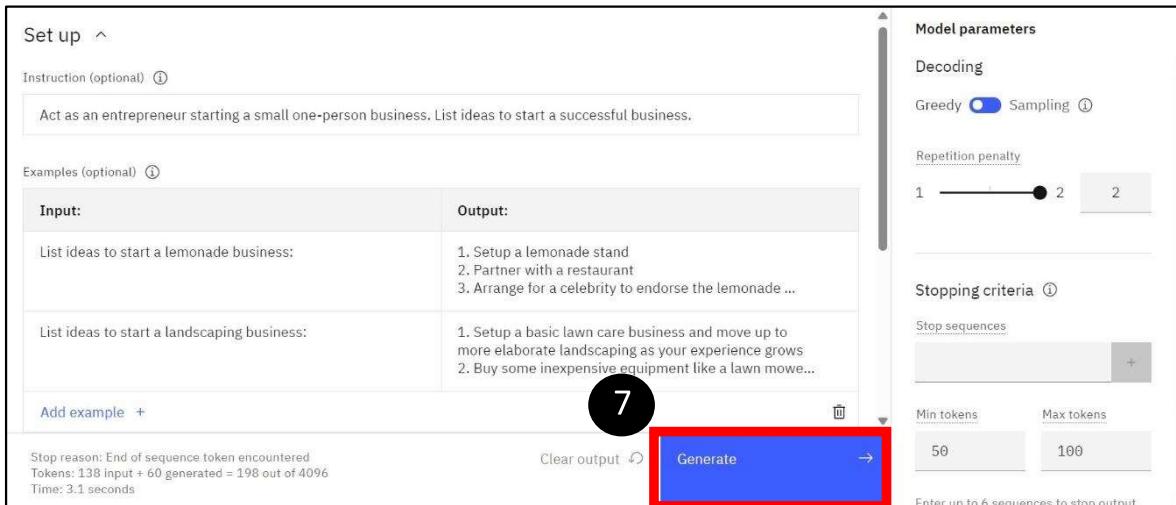
Stop sequences

Min tokens 50 Max tokens 100

Enter up to 6 sequences to stop output

7

Clear output ↻ Generate →



The results are now from the perspective of an entrepreneur starting a small one-person business. It contains some innovative ideas and practical advice for starting a dog-walking business.

Try ^

Test your prompt ⓘ

Input:

List ideas to start a dog-walking business:  
1.

Output:

Start by walking dogs in your neighborhood. 2.  
Advertise your services on local bulletin boards and through word of mouth. 3. Get a business license and insurance. 4. Consider offering other services, such as pet sitting or dog training. 5. Set up a website to advertise your services.

AI

Stop reason: End of sequence token encountered  
Tokens: 161 input + 58 generated = 219 out of 4096  
Time: 2.3 seconds

Clear output ↻

Generate →

8. It is always a good idea to test your prompt on multiple examples before you assume that the prompt is working as intended. For this reason, you decide to give the foundation model one more prompt to test its effectiveness in teasing out responses from the foundation model. You decide to prompt the foundation model to list ideas to start a car detailing business.

Type in the following prompt in the **Input:** text box under the **Try** section:

**List ideas to start a car detailing business:**

1.

The screenshot shows the 'Try' interface with a red box highlighting the 'Input:' field. Inside the field, the text 'List ideas to start a car detailing business:' is entered. A large black circle with the number '8' is overlaid on the right side of the interface, pointing to the AI-generated output. The output text reads: 'Start by walking dogs in your neighborhood. 2. Advertise your services on local bulletin boards and through word of mouth. 3. Get a business license and insurance. 4. Consider offering other services, such as pet sitting or dog training. 5. Set up a website to showcase your services.' Below the input field, there is a 'New test +' button. At the bottom, status information includes 'Stop reason: End of sequence token encountered', 'Tokens: 161 input + 58 generated = 219 out of 4096', and 'Time: 2.9 seconds'. On the right, there are sections for 'Stopping criteria' (with 'Min tokens' at 50 and 'Max tokens' at 100) and a 'Repetition penalty' slider set to 2. A 'Clear output' button and a blue 'Generate' button are also visible.

9. Clear the output (next to the Generate button) and click the blue **Generate** button.

The screenshot shows the 'Try' interface with a red box highlighting the 'Output:' field. Inside the field, the text 'Generated output appears here.' is displayed. A large black circle with the number '9' is overlaid on the right side of the interface, pointing to the blue 'Generate' button. The 'Generate' button is highlighted with a red box. Below the input field, there is a 'New test +' button. At the bottom, status information includes 'Stop reason: End of sequence token encountered', 'Tokens: 161 input + 58 generated = 219 out of 4096', and 'Time: 2.3 seconds'. On the right, there are sections for 'Repetition penalty' (set to 2), 'Stopping criteria' (with 'Min tokens' at 50 and 'Max tokens' at 100), and a note 'Enter up to 6 sequences to stop output'.

The response is appropriate, and you are delighted with your progress.

Try ^

Test your prompt. ⓘ

Input:	Output:
List ideas to start a car detailing business: 1.	Buy a car detailing kit from an auto store. 2. Get a business license and insurance. 3. Advertise your services in the local paper. 4. Purchase a vehicle to use for your business. 5. Set up a website. 6. Get a mobile phone number.

New test +

Stop reason: End of sequence token encountered  
Tokens: 159 input + 56 generated = 215 out of 4096  
Time: 2.3 seconds

Clear output ⌂ Generate →

This screenshot shows the AI interface's Freeform tab. In the input field, the user has typed a prompt asking for ideas to start a car detailing business, including a numbered example. The output field displays a detailed list of steps, starting with buying a kit and getting a license. Below the input and output fields, there is a status bar showing the stop reason (end of sequence token), tokens used (159 input + 56 generated = 215 out of 4096), and time taken (2.3 seconds). At the bottom right, there are buttons for 'Clear output' and 'Generate'.

10. At the beginning of the prompt engineering exercise, you saw that there are three ways to construct your prompt: using the Structured or the Freeform interface as well as using the Chat interface. You have so far focused on the Structured way of creating a prompt and now it is time to explore the Freeform tab. Select the **Freeform** tab to switch interfaces.

Chat Structured **Freeform**

Set up ^ 10

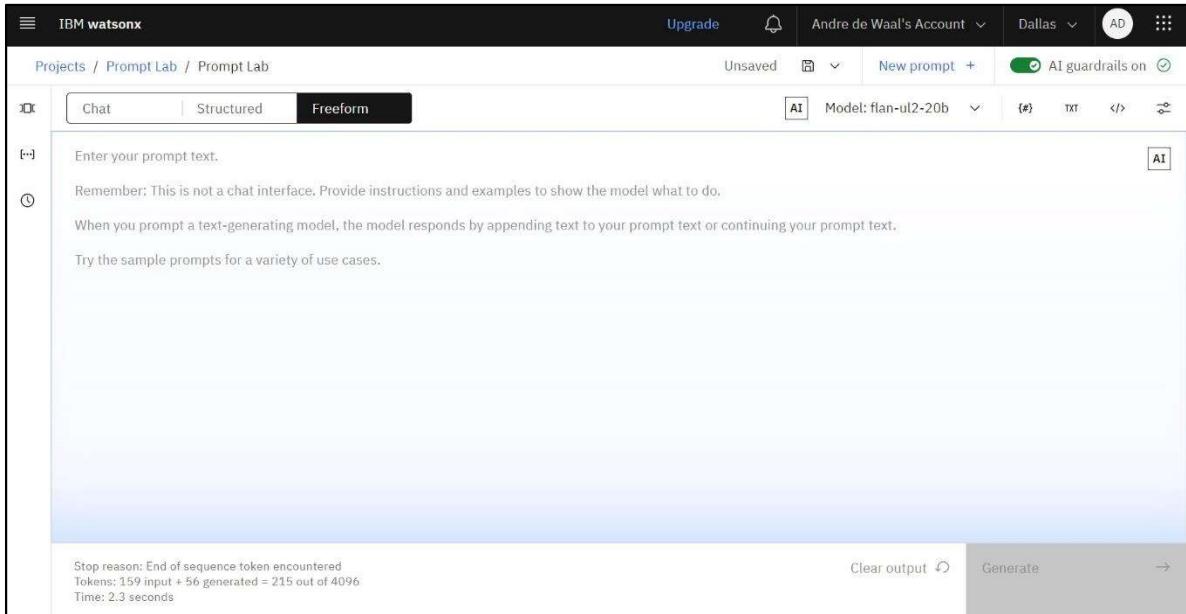
Instruction (optional) ⓘ

Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.

AI Model: flan-ul2-20b Model parameters  
Decoding Greedy Sampling ⓘ

This screenshot shows the AI interface with the 'Freeform' tab highlighted by a red box. The main input field contains a prompt for generating ideas for starting a small one-person business. On the right side, there is a sidebar titled 'Model parameters' with options for 'Decoding' (set to 'Greedy') and 'Sampling'. The number '10' is prominently displayed in a black circle in the center of the interface.

The interface changes and you can provide your prompt in freeform format containing instructions as well as examples that you previously entered in the Structured interface.



11. Select the **Structured** Interface.
12. Select the **TXT** tab in the upper right quadrant of the interface to view the full prompt text.



13. The panel contains the prompt in text format. Copy the full prompt text to the clipboard by clicking on the **Copy to clipboard** button.

The screenshot shows the IBM Watsonx interface with the 'Prompt Lab' selected. The 'Structured' tab is active. In the main area, there's an 'Instruction (optional)' field containing the text: 'Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.' Below it is an 'Examples (optional)' section with two rows. The first row shows 'Input: List ideas to start a lemonade business:' and 'Output: 1. Setup a lemonade stand 2. Partner with a restaurant 3. Arrange for a celebrity to endorse the lemonade ...'. The second row shows 'Input: List ideas to start a landscaping business:' and 'Output: 1. Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows 2. Buy some inexpensive equipment like a lawn mow...'. At the bottom, there's a 'Generate' button. To the right, a sidebar shows expanded examples for both business types. A red box highlights the 'View full prompt text' button, which is circled with the number 13.

14. Switch to the Freeform interface and paste the **full prompt text** into the Freeform interface.

The screenshot shows the IBM Watsonx interface with the 'Prompt Lab' selected. The 'Freeform' tab is active. The main input area contains the full prompt text: 'Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.' Below this, there are three sections of examples, each with an 'Input:' and 'Output:' pair. A large red box highlights the entire input area. To the right, there's a sidebar with a cube icon and the text 'Enter a test prompt to see the raw prompt'. At the bottom, there's a 'Generate' button, which is circled with the number 14.

15. Press **Enter** to confirm that your prompt is accepted.

The **Generate** button becomes active.

The screenshot shows the IBM watsonx interface with the 'Freeform' tab selected. A prompt is entered: "Act as an entrepreneur starting a small one-person business. List ideas to start a successful business." Below this, three examples of generated output are shown:

- Input: List ideas to start a lemonade business:  
Output: 1. Setup a lemonade stand  
2. Partner with a restaurant  
3. Arrange for a celebrity to endorse the lemonade stand
- Input: List ideas to start a landscaping business:  
Output: 1. Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows  
2. Buy some inexpensive equipment like a lawn mower, rakes, shovels, etc.  
3. Purchase used lawn furniture from estate sales or yard sales at very low cost
- Input: List ideas to start a car detailing business:  
Output: 1.

At the bottom, a message says "Stop reason: End of sequence token encountered" and "Tokens: 159 input + 56 generated = 215 out of 4096 Time: 2.3 seconds". There are "Clear output" and "Generate" buttons, with "Generate" being blue.

16. Click the blue **Generate** button.

The screenshot shows the same interface as above, but with a large black circle containing the number "16" positioned over the "Generate" button. The "Generate" button is highlighted with a red rectangle.

The response is very similar to what was generated using the Structured interface. Due to the non-deterministic nature of LLMs, your output may not always be the same as what is given in this guide.

The screenshot shows the IBM Watsonx Prompt Lab interface. The top navigation bar includes 'Upgrade', 'Andre de Waal's Account', 'Dallas', and a 'Projects / Prompt Lab / Prompt Lab' section. The main area has tabs for 'Chat', 'Structured', and 'Freeform', with 'Freeform' selected. A sidebar on the right shows the full prompt text: 'Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.' Below this, three examples are listed: 1. Lemonade business (Setup stand, partner with restaurant, arrange celebrity endorsement), 2. Landscaping business (Setup lawn care business, buy equipment, purchase furniture), and 3. Car detailing business (Buy kit, get license, bank account, credit card, website, business cards). At the bottom, there are buttons for 'Clear output', 'Generate', and a progress bar indicating the process is complete.

After you become familiar with the Prompt Lab, it might be easier, faster, and as you can see more convenient to construct the prompt in the unstructured or freeform way. The response from the foundation model is unaffected by which interface you prefer, but the response may be affected by your prompt lab settings.

- If you would like to generate alternative responses, change the **Decoding** parameter from **Greedy** to **Sampling**. Selecting **Sampling** instructs the foundation model to customize the variability of word selection. This setting change lets the model create alternative responses instead of re-creating a previous response. Select the **Model parameters** tab and select **Sampling** under the **Decoding** section.

The screenshot shows the same Prompt Lab interface as above, but with the 'Model parameters' tab selected in the top navigation. A large red box highlights the 'Decoding' section, where the 'Sampling' button is selected instead of 'Greedy'. To the left of the red box, a large black circle contains the number '17', indicating this is step 17. The rest of the interface remains the same, showing the generated responses for the three business types.

18. Scroll down in the Model parameters panel and change the **Min tokens** to **100** and the **Max tokens** to **200**.
19. Clear the output and click the blue **Generate** button.

IBM watsonx

Projects / Prompt Lab / Prompt Lab

Chat | Structured | Freeform

AI Model: flan-ul2-20b

Model parameters

Decoding: Greedy (Sampling)

Temperature: 0.7

Top P (nucleus sampling): 1

Top K: 100

Random seed:

Repetition penalty: 50

Stop reason: End of sequence token encountered  
Tokens: 159 input + 80 generated = 239 out of 4096  
Time: 15.7 seconds

Clear output | Generate →

The LLM's completion response is good given the short instruction and two examples.

IBM watsonx

Projects / Prompt Lab / Prompt Lab

Chat | Structured | Freeform

AI Model: flan-ul2-20b

Model parameters

Decoding: Greedy (Sampling)

Temperature: 1

Top P (nucleus sampling): 1

Top K: 100

Random seed:

Repetition penalty: 50

Stopping criteria

Stop sequences:

Min tokens: 100 | Max tokens: 200

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

Reset to default

Stop reason: End of sequence token encountered  
Tokens: 159 input + 107 generated = 266 out of 4096 | Seed: 982735830  
Time: 6 seconds

Clear output | Generate →

**Important:** After you set the decoding parameter to **Sampling**, your responses might be different from what is given in this course. This result is as expected as more variability in word selection might lead to different responses. This non-deterministic behavior is inherent in LLMs.

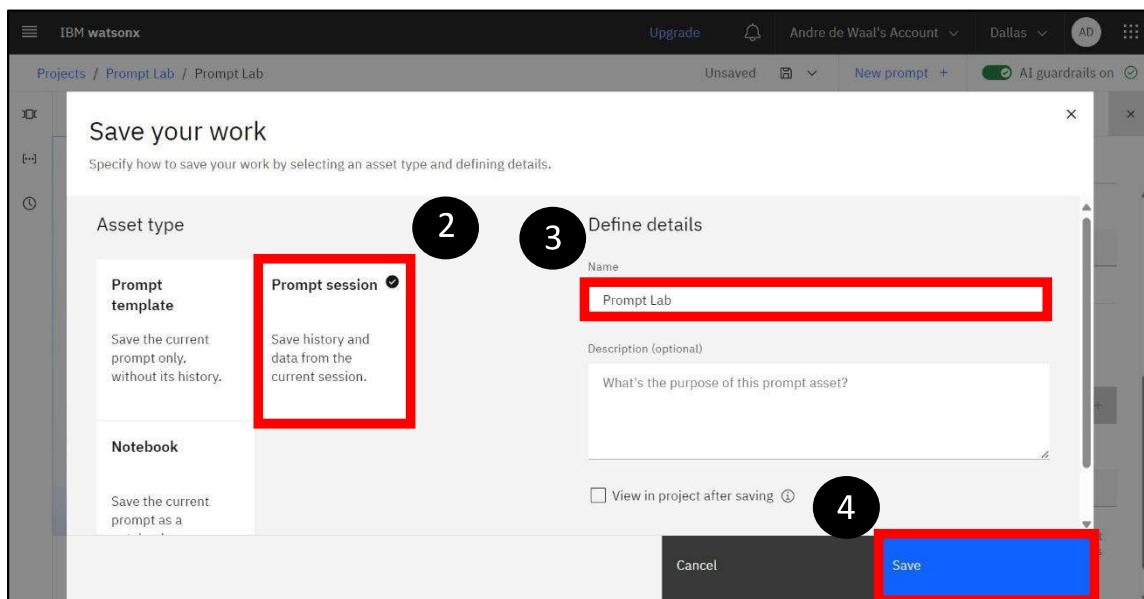
## Section 3: Inspect the foundation model Python code

You are also interested in seeing the Python code that was generated in the background. By using WatsonX Studio to interact with the LLM.

1. Expand the **Save menu** expander arrow (disk drive) and select the **Save as** in the upper-right quadrant of the interface.



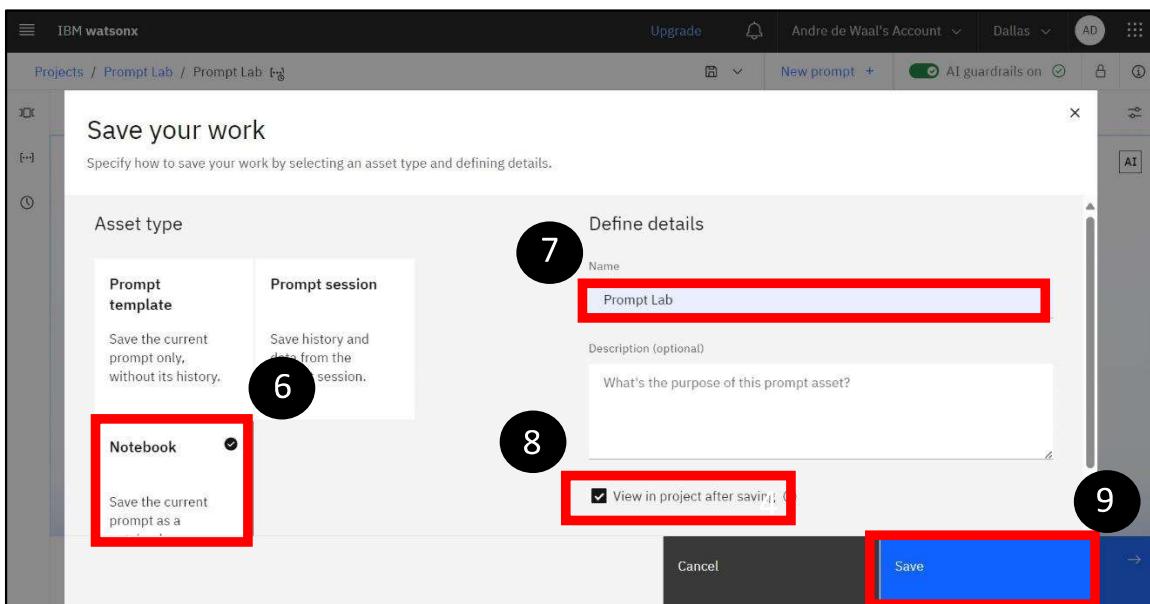
2. Select the **Prompt session** tile.
3. In the **Name** field give the session a descriptive name such as **Prompt Lab**.
4. Click **Save** to save the Prompt Session.



5. Expand the **Save menu** expander arrow (disk drive) and select the **Save as** in the upper-right quadrant of the interface.



6. Select the **Notebook** tile.
7. In the **Name** field give the notebook a descriptive name such as **Prompt Lab**.
8. Select the **View in project after saving** check box.
9. Click **Save** to save the Prompt Session.



The Notebook opens and you can inspect the code, modify the code, and use the notebook as a template for other Prompt Lab experiments.

The screenshot shows the IBM Watsonx interface with the title "Prompt Lab" and subtitle "Part of IBM watsonx.ai®". The main content area is titled "Prompt Notebook - Prompt Lab Notebook v1.1.0". It contains a note about inferencing steps and Python API commands for authentication. A "Note" section states that code generated by Prompt Lab will execute successfully if modified or reordered. Below is a "Notebook goals" section listing learning objectives related to Python functions and Model objects.

10. Scroll down to the **Defining the inferencing input** section (cell 6) of the Jupyter Notebook to see the Python prompt equivalent of what you input as text in the Prompt Lab in watsonx.ai Studio.

```
In [ ]: prompt_input = """Act as an entrepreneur starting a small one-person business. List ideas to start a successful bus
Input: List ideas to start a lemonade business:
Output: 1. Setup a lemonade stand
2. Partner with a restaurant
3. Arrange for a celebrity to endorse the lemonade stand

Input: List ideas to start a landscaping business:
Output: 1. Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows
2. Buy some inexpensive equipment like a lawn mower, rakes, shovels, etc.
3. Purchase used lawn furniture from estate sales or yard sales at very low cost

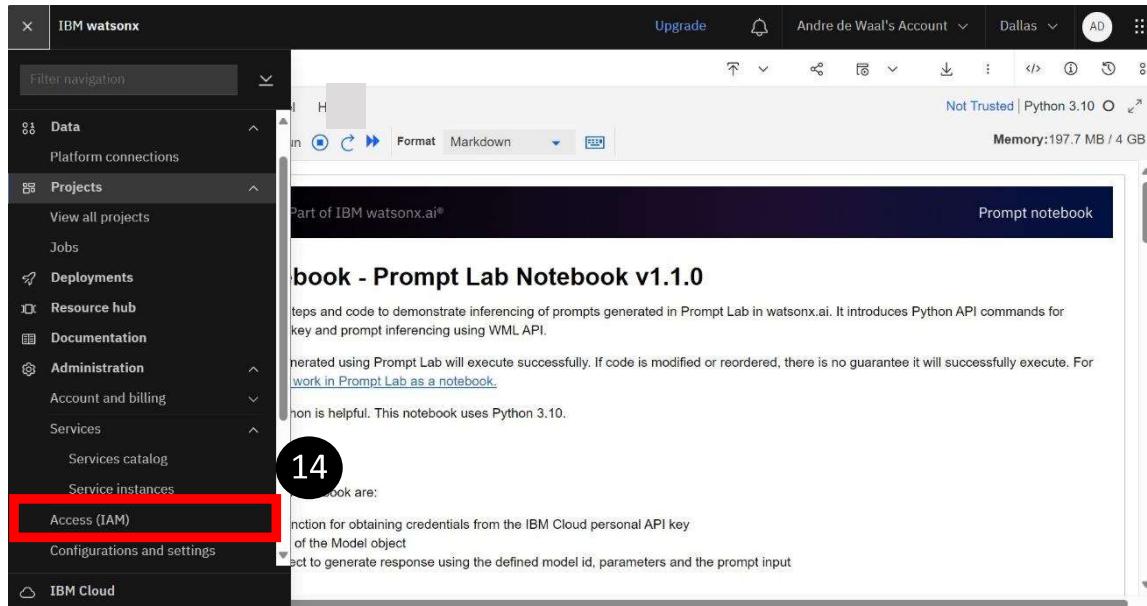
Input: List ideas to start a car detailing business:
1.
Output:
"""
```

11. Click the **Edit** icon (pencil) in the upper-right quadrant of the interface.

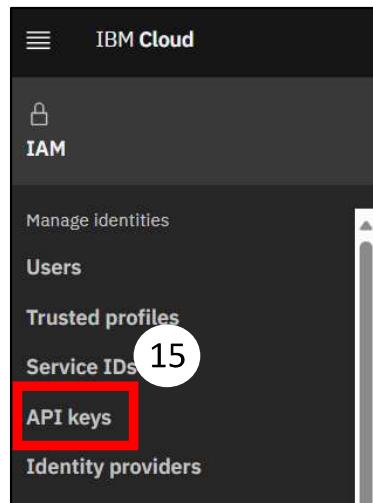


The runtime for the Prompt Lab Jupyter Notebook instantiates (stands up).

12. **Retrieve** your API key (for more information on how to generate your IBM Cloud personal API key, click the [Documentation](#) link).
13. Click the **Navigation Menu** (the four horizontal bars also known as the hamburger menu) in the upper-left quadrant of the screen.
14. Select **Access (IAM)** in the left panel.



15. Select **API keys** from the menu on the left.



16. Click **Create**.

## API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. [Learn more.](#)

Looking for more options to manage API Keys? Try IBM Cloud® Secrets Manager for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

View: My IBM Cloud API keys ▾

API keys associated with a user's identity have the same access that the user is assigned across all accounts. To update the access for an API key, assign or remove access for the user.

16

Create +

Status	Name	Description	Date created
	No API keys	Looks like there aren't any API keys. Click <a href="#">Create</a> to get started.	

17. Enter a **name** and **description** in the corresponding fields, for example as **watsonx** and **watsonx API key**.

18. Click **Create**. The API key is generated.

17

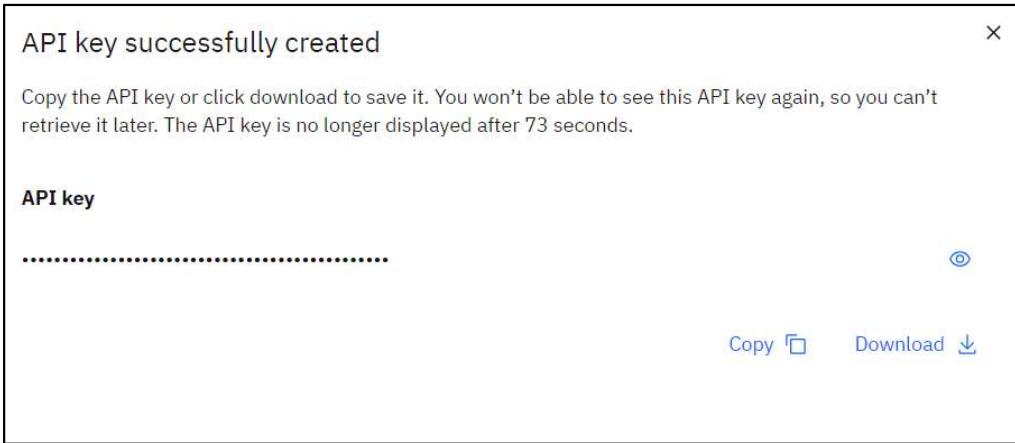
Create IBM Cloud API key

Name  
watsonx

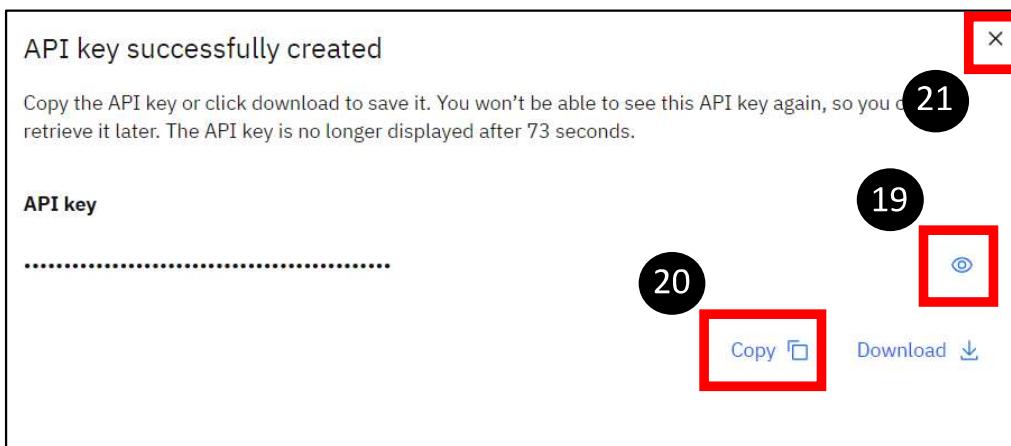
Description  
watsonx API key

18

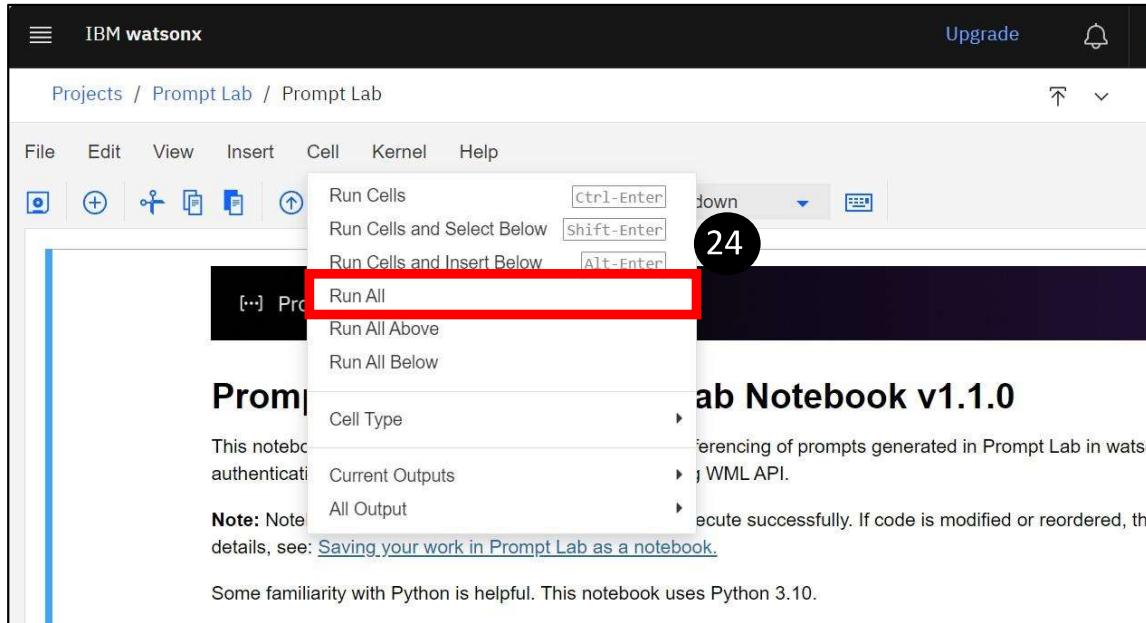
Cancel Create



19. Click the **Show** (eye icon) on the far right to display your API key.
20. Click **Copy** to copy your API key. It is a good idea to save your API key as you may need it for future use.
21. Press **x** to close the dialog box.



22. Click the **Prompt Lab --- Prompt Lab | IBMwatson** tab in your browser to return to the watsonx GUI and your Jupyter Notebook.
23. Click the **edit** icon (underlined pencil) in the upper-right quadrant of your screen to Instantiate the Python runtime for the Prompt Lab.
24. Click **Cell** to expand the Cell menu and then select **Run All** to run the notebook.



25. Paste your API key in the browser when asked and press **Enter**.
26. The rest of the Notebook is run, and the following response is generated.

### Execution

Let us now use the defined Model object and pair it with input and generate the response:

```
In [7]: ⏪ print("Submitting generation request...")
generated_response = model.generate_text(prompt=prompt_input, guardrails=True)
print(generated_response)
```

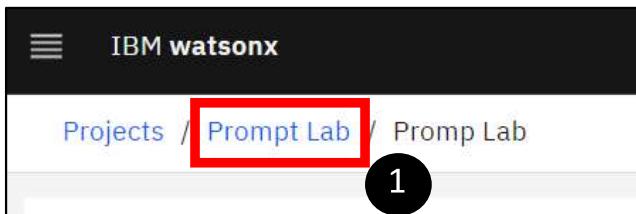
```
Submitting generation request...
Locate a building, workshop or garage to use as your headquarters. 2. Purchase or build the car wash equipment and supplies you need. 3. Advertise your services to car owners and businesses. 4. Get a license and insurance. 5. Decide whether you will do mobile or stationary car washes. 6. Plan and start your business. 7. Start detailing cars and building your clientele. 8. Renovate your shop. 9. Start hiring employees. 10. Invest in marketing. 11. Set up an app for an online presence.
```

You might notice that the response is slightly different from what you got previously as the **Decoding** parameter is set to **Sampling**. The response is acceptable.

## Section 4: Model options and tokens

Different foundation models might also generate different responses to a prompt. This difference is expected as the models might have different architectures, a different number of parameters, and the models might have been trained on different data sets.

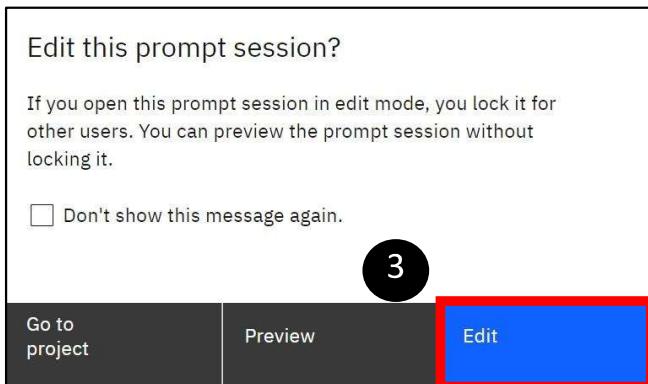
1. Click the **Prompt Lab** in the upper left quadrant of your project to get back to your Prompt Lab Project.



2. Click on the **Prompt Lab** **Prompt Session** under the **Assets** tab to select your Prompt Session.

Name	2	Last modified	↓
... Prompt Lab		Now Modified by you	:

3. Click **Edit** to edit the Prompt Session



You returned to your **Freeform** prompt session.

4. Click the current foundation model **flan-ul2-20b** and select **View all foundation models** to see a list of the currently available foundation models.

Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.

**Input:** List ideas to start a lemonade business:  
**Output:**

1. Setup a lemonade stand
2. Partner with a restaurant
3. Arrange for a celebrity to endorse the lemonade stand

**Input:** List ideas to start a landscaping business:  
**Output:**

1. Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows
2. Buy some inexpensive equipment like a lawn mower, rakes, shovels, etc.
3. Purchase used lawn furniture from estate sales or yard sales at very low cost

**Input:** List ideas to start a car detailing business:  
**Output:**

1. Choose a date to get your business started and then begin advertising, get your website going and get your business cards made.
2. Start by detailing cars for friends and family to build up a clientele base.
3. Use your previous experience to help you get started, especially if you have experience in the automotive field.
4. Create an ad on the internet and in local newspapers to let people know you are available.
5. Take the time to learn how to properly detail a car.

Clear output Generate

The following models are available (this list changes over time and new models are added and removed from watsonx.ai).

Select a foundation model

Select a model that best fits your needs. All models support English text. Check the model information for other supported languages.

Search for a model or task

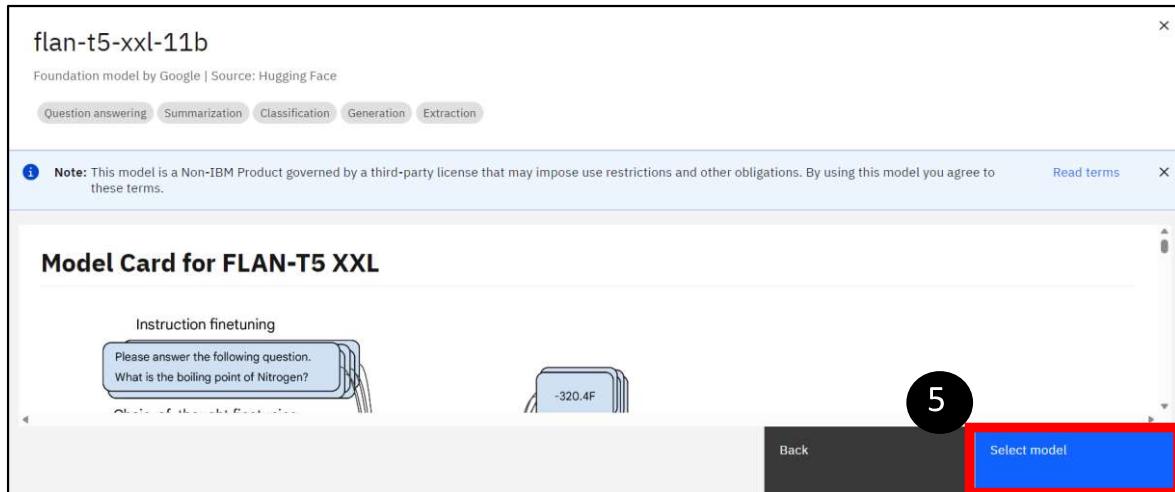
<b>granite-13b-chat-v2</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: InstructLab	<b>mt0-xxl-13b</b> An instruction-tuned iteration on mt6, which are particularly well-suited for... Provider: BigScience Type: Provided mo...	<b>codellama-34b-instruct-hf</b> Code Llama is an AI model built on top of Llama 2, fine-tuned for generating and discussing code. Provider: CodeLlama Type: Provided mo...	<b>flan-t5-xl-3b</b> A pre-trained T5—an encoder-decoder model pre-trained on a mixture of supervised / unsupervised tasks converted into a sequence-to-sequence model. Provider: Google Type: Provided mo...	<b>flan-t5-xxl-11b</b> Flan-T5 is an 11 billion parameter model based on the Flan-T5 family. Provider: Google Type: Provided mo...	<b>flan-ul2-20b</b> Flan-U2 is an encoder-decoder model based on the T5 architecture and instruction-tuned using the Fine-tuned... Provider: Google Type: Provided mo...
<b>merlinite-7b</b> Merlinite-7b is a Mistral-7b-derivative model trained with the LAB methodology, using Mistral-8x7b-Instruct as a teacher... Provider: Mistral AI, tuned... Type: InstructLab	<b>mixtral-8x7b-instruct-v0...</b> Mixtral-8x7b-instruct-v0.1-gptq model is made with AutoGPTQ, which mainly leverages the quantization technique to... Provider: Mistral AI, tuned... Type: Provided mo...	<b>granite-13b-instruct-v2</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: Provided mo...	<b>granite-20b-code-instruct</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: Provided mo...	<b>granite-20b-multilingual</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: InstructLab	<b>granite-34b-code-instruct</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: Provided mo...
<b>granite-3b-code-instruct</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: Provided mo...	<b>granite-7b-lab</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: InstructLab	<b>granite-8b-code-instruct</b> The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for... Provider: IBM Type: Provided mo...	<b>llama-2-13b-chat</b> Llama-2-13b-chat is an auto-regressive language model that uses an optimized transformer architecture. Provider: Meta Type: Provided mo...	<b>llama-2-70b-chat</b> Llama-2-70b-chat is an auto-regressive language model that uses an optimized transformer architecture. Provider: Meta Type: Provided mo...	<b>llama-3-70b-instruct</b> Llama-3-70b-instruct is an auto-regressive language model that uses an optimized transformer architecture. Provider: Meta Type: Provided mo...

For example, **Flan-ul2-20b** is a general foundation model with 20 billion parameters and it is suitable for zero-shot and few-shot prompt engineering in watsonx.ai's Prompt Lab.

Sometimes, you might also need to experiment with a more complex model if the response to your prompt is unsatisfactory. You might also need to explore different types of models based on different architectures, such as Encoder-Decoder or Decoder-only models (these are types of transformers, the technology behind foundation models).

Models might also be trained on different data sets, and in different languages. Inspect the documentation before selecting a particular model.

5. In many cases you can get similar results that use a smaller model (which might save resources). Click on the **flan-t5-xxl-11b** tile to select a smaller model (11 billion parameters), still based on the same underlying model family (Flan-T5). Read the model card, then click **Select model**.



6. Navigate back to the **Prompt Lab** by selecting the **Prompt Lab Prompt Session** asset from the **Assets** tab.

A screenshot of the IBM WatsonX interface, specifically the "Assets" tab under the "Prompt Lab" project. The "Assets" tab is highlighted with a blue underline. Below it, there's a search bar with "Find assets" and a "New asset" button. A table lists "5 assets" under the "All assets" section. One asset, "Prompt Lab" (a "Prompt session"), is selected and highlighted with a red rectangular box. A large black circle with the number "6" is overlaid on the "Prompt Lab" row. The table columns include "Name", "Last modified", and a more options menu.

7. Click **Edit** to edit the Prompt Session. You are back at your Prompt Session.
8. Remove the previous response from the Prompt Lab and click **Generate** to generate a new response.

The screenshot shows a user interface for generating AI prompts. At the top, there are tabs for Chat, Structured, and Freeform, with Freeform selected. The model used is flan-t5-xxl-11b. Below the tabs, there are three examples of prompts:

- Act as an entrepreneur starting a small one-person business. List ideas to start a successful business.**  
Input: List ideas to start a lemonade business:  
Output: 1. Setup a lemonade stand  
2. Partner with a restaurant  
3. Arrange for a celebrity to endorse the lemonade stand
- Input: List ideas to start a landscaping business:**  
Output: 1. Setup a basic lawn care business and move up to more elaborate landscaping as your experience grows  
2. Buy some inexpensive equipment like a lawn mower, rakes, shovels, etc.  
3. Purchase used lawn furniture from estate sales or yard sales at very low cost
- Input: List ideas to start a car detailing business:**  
1.

At the bottom right, there are buttons for "Clear output" and "Generate". A large blue button labeled "Generate" is highlighted with a red border. A black circle with the number "8" is positioned above the "Generate" button.

The LLM generates the response using the smaller model. The response is not a lot different from that generated by the larger model and may be acceptable for your use cases. You can do some more experiments to evaluate this model for future use.

This screenshot continues the AI session from the previous one. The interface remains the same, with the Freeform tab selected and the flan-t5-xxl-11b model chosen. The user has added a new input line:

Input: List ideas to start a car detailing business:  
1.

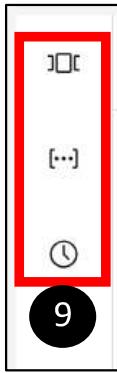
Output: Have specialized detailing equipment such as hot water pressure washers, brushless sanding belts, and a clay bar. 2. Offer a service that is specialized. e.g., detailing rims and tires, cleaning interiors and exteriors, etc. 3. Offer a variety of services. e.g., waxing, clayling, polishing, interior cleaning, etc. 4. Be fair and honest. 5. Get permits for your business. 6. Advertise in local newspapers. 7. Use a quality paint protectant that protects the paint as well as preventing oxidation. 8. Ask your customers for referrals

At the bottom, there is additional information about the generation process:

Stop reason: End of sequence token encountered  
Tokens: 157 input + 140 generated = 297 out of 4096 | Seed: 42841150  
Time: 4 seconds

The "Generate" button is again highlighted with a red border.

9. Click on the **Sample prompts**, **Saved prompts**, and **History icons** in the left panel of your prompt session to get additional information on tasks you can perform with suggested prompts, saved prompts, and a history of your current prompt session.



The **Sample prompts** might be useful for suggesting more tasks that a foundation model might solve, and how to get started with a particular task.

Category	Task	Description
Summarization	Meeting transcript summary	Summarize the discussion from a meeting transcript.
	Earnings call summary	Summarize financial highlights from a quarterly earnings call.
Classification	Scenario classification	Classify scenario based on project categories.
	Feedback classification	Classify feedback about insurance customer service.
Generation	Marketing email generation	Generate email for marketing campaign.

The last task for you is to recommend to your manager how this new technology might be used in their company. This recommendation is covered in the next exercise.

End of exercise

### 3. Exercise review and wrap-up

In the exercises, you used the functionality available in IBM watsonx to build prompts in the Prompt Lab. You started with a basic prompt (zero-shot prompting) and iterated until you had a prompt that performed well on a couple of examples. During each successive iteration, you improved the cue, added more examples, and finally added an imperative statement (called an instruction) to improve your prompt. You also looked at the Prompt Lab settings and adjusted several settings to steer the foundation model toward generating the wanted responses.

You took a journey to explore foundation models in the watsonx platform. The only question remaining is how your knowledge can benefit your company in their quest to use the power of foundation models.

After careful consideration, you identified at least four use cases where foundation models can possibly be used successfully in the company:

1. Generating marketing emails to customers.
2. Answering health product-related questions.
3. Doing sentiment analysis of survey responses.
4. Classifying survey responses into different categories, such as product, price, quality, etc., for further action.

Now you will prepare a presentation on his findings and share it with colleagues and management.

After completing these exercises, you should have all the necessary skills to use the Prompt Lab successfully in watsonx.