

IBM Bob Hackathon



IBM Bob Hackathon

IBM Bob is an intelligent software development partner designed to support teams throughout the journey from idea to solution. Rather than focusing on individual coding tasks, Bob understands the full context of a project, including goals, existing code, and development intent. This enables users to move faster, make well informed decisions, and focus on solving meaningful problems instead of getting blocked by complexity or repetitive work.

Bob works as a collaborative partner that helps translate ideas into high quality software outcomes. By assisting with understanding code, organizing solution structure, generating documentation and tests, and automating routine development tasks, Bob makes it easier for builders at any skill level to contribute, experiment, and deliver impactful solutions with confidence.

This guide explains how to access IBM Bob for the purposes of this hackathon, explore optional IBM watsonx technologies, and use them together to build a proof-of-concept solution for the hackathon.

Contents

- The hackathon expectation 4
 - A note on using other technologies..... 4
 - A note on data sets before you begin..... 4
- Get started with IBM Bob 5
 - Confirm your hackathon IBM Bob account..... 5
 - Bobcoins..... 6
 - Working with IBM Bob..... 8
 - 1. Bob IDE (Required)..... 8
 - Install Bob IDE 8
 - Sign into Bob IDE..... 8
 - Best practices and FAQ 9
 - Bob IDE features and configuration 10
 - Bob IDE hands-on exercises 15
 - 2. Bob Shell (Optional)..... 17
 - Install Bob Shell 17
 - Authentication 17
 - Start an interactive session 17
 - Start a non-interactive session 17
 - Bob Shell configuration 17

- Bob Shell usage examples 18
- Exporting Bob task session reports for judging..... 18
- Get started with IBM watsonx (Optional) 20
 - Note on available services..... 20
 - Note on preventing exposure of IBM Cloud credentials 20
 - Note on IBM Cloud service usage 21
- Request your hackathon IBM Cloud account..... 21
- Accessing your hackathon IBM Cloud account..... 22
- Accessing and utilizing IBM watsonx products..... 24
- 1. IBM watsonx Orchestrate..... 25
 - Accessing watsonx Orchestrate 25
 - Discover the catalog 26
 - Building agents 27
 - Using agents in the chat 33
 - Managing app connections 33
 - watsonx Orchestrate API 34
 - Building AI assistants 34
- Quick start hands-on exercises..... 34
- 2. IBM watsonx.ai..... 36
 - Note on IBM watsonx.ai service usage 36
 - Note on available watsonx.ai capabilities 37
 - Access Prompt Lab on watsonx.ai 37
 - Work with the watsonx.ai Prompt Lab 38
 - Prompt Lab editor..... 39
 - Selecting an AI model 39
 - Programmatic access (API/SDK) 40
 - watsonx.ai AI agent libraries and tutorials 43
 - Quick start hands-on exercises..... 44
 - Save your Prompt Lab session 45
 - Save your work on watsonx.ai 46
- Appendix: Example use cases 47

The hackathon expectation

In this hackathon, you will design and build a **proof-of-concept solution** using **IBM Bob**, aligned to the following theme:

Turn idea into impact faster

Got an idea for how to work smarter and faster? Create a solution that speeds the way you work every day. Maybe you need to get up to speed on existing code quickly, or generate documentation and tests, or reduce effort spent on repetitive tasks that slow your team down.

Use IBM Bob as your intelligent development partner to bring your idea to life. Bob understands intent, reads complete repository context, explains logic with clarity, automates complex transformations, and streamlines multi-step work. Show how your solution, powered by Bob, can help builders at any skill level deliver high quality software with greater efficiency and confidence.

Refer to [example use cases](#) to help you brainstorm ideas for building a solution.

Important: Participants are required to export and upload **Bob IDE task session report** in their code repository for judging purposes. Refer to the [instructions on how to export task session reports](#).

Participants may also **optionally** use the following watsonx products:

- [IBM watsonx Orchestrate](#) – a no-code and low-code platform designed to orchestrate AI agents across business workflows. With watsonx Orchestrate, you can create, deploy, and manage intelligent agents and assistants that automate tasks, streamline processes, and enhance productivity.
- [IBM watsonx.ai](#) – a powerful AI studio that supports the development of agentic AI solutions using a wide range of foundation models. Through the Prompt Lab, participants can experiment with IBM's Granite models and other leading models to create agents that understand and respond to natural language. IBM watsonx.ai can also serve as an inference provider for your agents, allowing them to generate responses, make decisions, and interact with users or systems intelligently.

A note on using other technologies

You may use any framework or technology to build your solution as long as you adhere to the product usage policies. However, to be eligible for judging, your solution must showcase **IBM Bob IDE** as a core component.

A note on data sets before you begin

Participants are required to bring their own datasets to build the solution aligning to your use case. As you collect data for your project, you'll want to use best practices. Here are helpful tips:

- Teams are responsible for ensuring data is compliant.
- Data from public websites may be used, if the terms allow for commercial use, but please keep a list of the websites you use.

- Do not use data or assets containing company confidential data, or any other data without permission from the data owner. Teams are responsible for getting approval.
- Do not use any client data.
- Do not use any data containing personal information (PI).
- Do not use data obtained from social media.

Get started with IBM Bob

To access and use the hackathon-provisioned IBM Bob account, participants must be registered for the hackathon.

Important: If you already have access to Bob through your personal account, please use the hackathon-provisioned Bob account only for this event and to avoid consuming usage on your personal account.

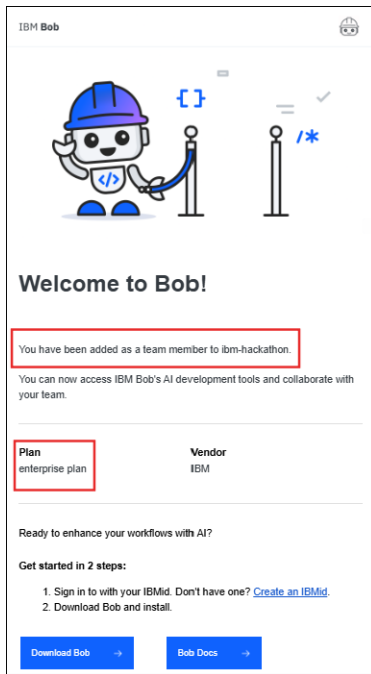
Confirm your hackathon IBM Bob account

At the start of the hackathon, all registered members will receive an email invite to join the **hackathon-provisioned Bob account**. Follow the steps below to confirm your account:

1. Check the email inbox you used to register for the hackathon and open the invitation email received from the IBM Bob team for accessing your hackathon-provisioned Bob account. To confirm that you have the correct email, look for the following details:
 - You have been added as a team member to **ibm-hackathon-xxx**
 - **Bob Plan:** Enterprise plan

Note: The team name mentioned above is only used to confirm that you are accessing the correct hackathon-provisioned Bob account. It is not related to your team formation for the hackathon.

Please check your Junk/Spam folders if you cannot find the email in your inbox. You can also quickly search for “IBM Bob” to locate the email.



2. [Create an IBMid](#) if you have not created one yet. IBMid is required to access Bob.
3. Next, you can:
 - [Install Bob and get started with solution building](#)
 - [Access Bob admin portal and view account information](#)

Bobcoins

Each interaction with Bob's AI capabilities consumes Bobcoins. For this hackathon, **40 Bobcoins** will be automatically applied to your hackathon-provisioned **IBM Bob account**. These Bobcoins are intended to be sufficient for designing and building a compelling proof-of-concept submission.

It is recommended that you plan ahead with your teammates to divide tasks so that you can make full use of the total Bobcoins allocated across all team members. Once your Bobcoin usage reaches **100% usage, no additional Bobcoins will be provided** as part of the hackathon. You may still continue working on your solution and final submission. For example, consider leveraging the optional watsonx technologies enabled for this hackathon.

Follow [Bob best practices](#) to use the tool effectively and efficiently.

You can monitor your Bobcoin usage in two ways:

- **Bob account**
 - Login to the [Bob admin portal](#).

Note: If you have **one or more personal Bob accounts**, please ensure to select the hackathon instance with the name **ibm-coding-challenge-xxx**. Ignore, if the hackathon-provisioned Bob

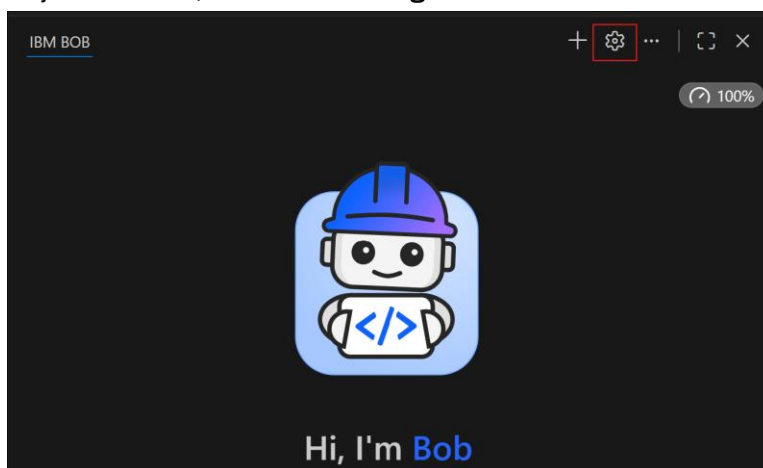
account is the only account associated with your registered email address.

- The admin page will display your Bobcoin usage in the Teams section. The team mentioned in your Bob account is not related to your team formation for the hackathon.

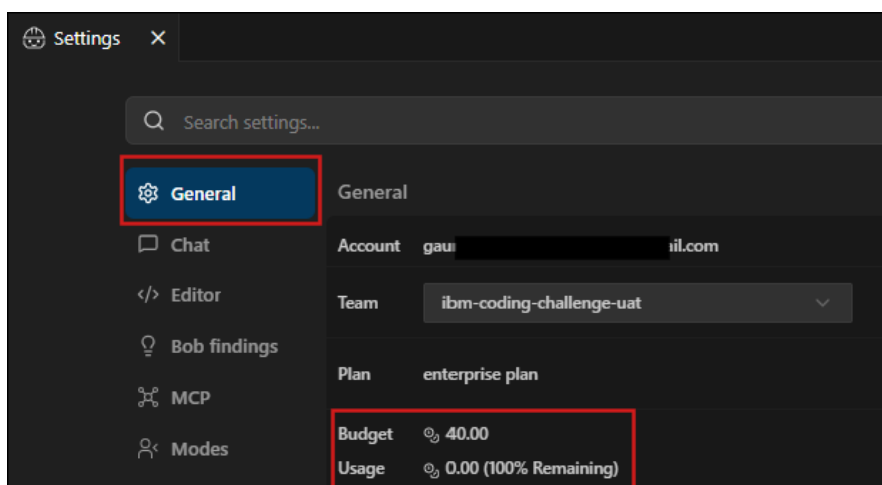
Teams			
Team Name	Spending limit	Usage	Remaining
ibm-hackathon	40 Bobcoins	0.00 Bobcoins	40.00 Bobcoins

- **Bob IDE**

- In your Bob IDE, select the **Settings** icon.



- The settings tab will open and under the **General** section, you can view your Bobcoin usage. If you have personal Bob accounts, please ensure to select the hackathon instance with the name **ibm-coding-challenge-xxx**.



Working with IBM Bob

After accessing your Bob account, you have the option to work with:

1. [Bob IDE](#) (Required)
2. [Bob Shell](#) (Optional)

Important: You must use Bob IDE for this hackathon. You will have to export the [Bob task session report for judging](#).

1. Bob IDE (Required)

Install the Bob IDE, sign into your Bob hackathon account, and start building your solution with Bob.

Install Bob IDE

Follow the [Bob IDE installation instructions](#).

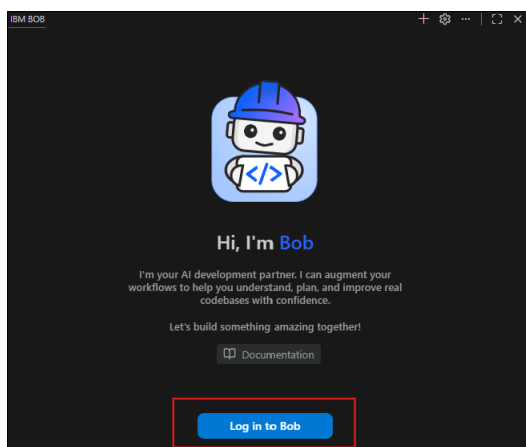
Note: Ensure that your system is compatible and meets the minimum system requirements to download and use Bob.

Sign into Bob IDE

To use Bob IDE after installing it, you have to sign in to Bob using your hackathon registration email.

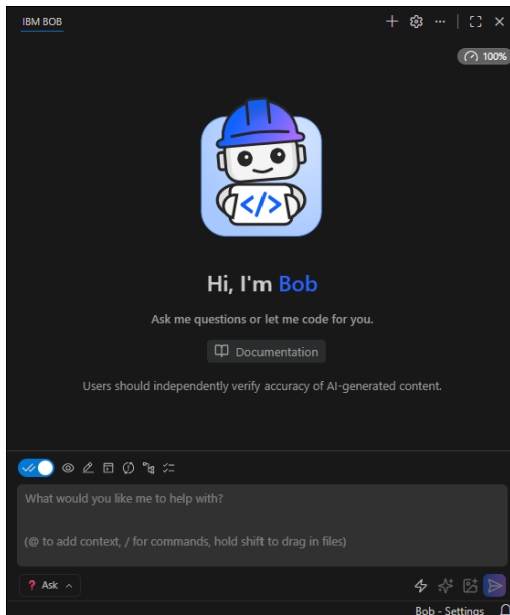
Important: An IBMid is required to authenticate. If you do not have one, see [Create an IBMid](#) to create one for your hackathon registered email.

- Open your Bob IDE and select the **Log in to Bob** button on the IDE.

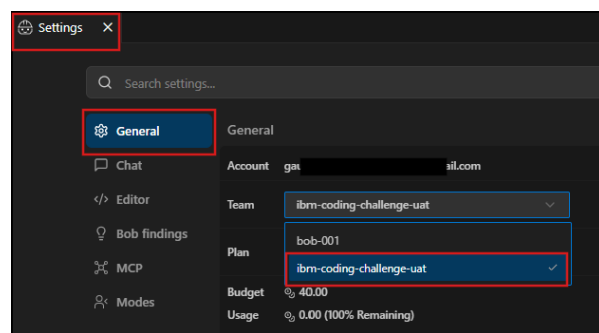
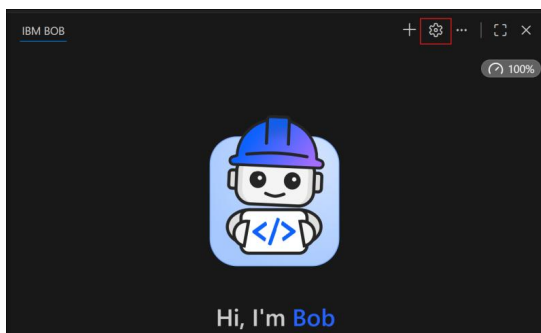


- Follow and complete the IBMid authentication process on your browser. Once authentication is

complete, Bob chat interface will be displayed for you to get started.



- Ignore this step if the hackathon-provisioned Bob account is the only account associated with your registered email address. If you have **one or more personal Bob accounts**, please ensure to select the hackathon instance with the name **ibm-coding-challenge-xxx** to avoid any usage on your personal account. To switch to hackathon-provisioned account, select the **Settings** icon in the Bob IDE and it will open the Settings tab. Under General, you can switch the team to **ibm-coding-challenge-xxx** instance.



Note: If you experience network issues with outbound traffic, you might need to configure your firewall settings. For more information, see [Configuring firewall rules for Bob](#).

Best practices and FAQ

Follow the [best practices guidelines](#) to get the most out of your experience with Bob.

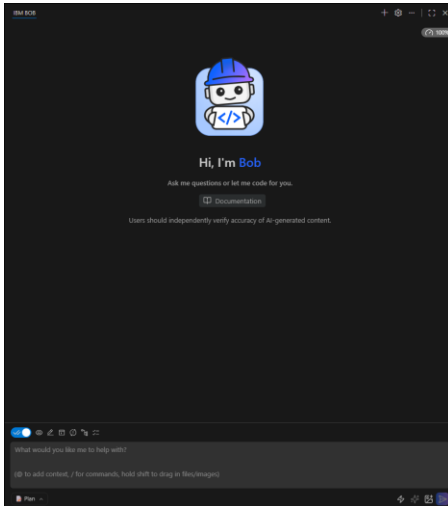
For common questions, troubleshooting tips, and additional guidance, refer to the [IBM Bob FAQ](#).

Bob IDE features and configuration

Explore Bob features and their configuration to enhance your experience working with Bob and solution building.

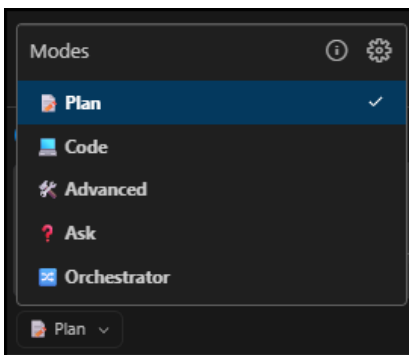
- [Chat interface](#)

The chat interface is your main way to interact with Bob. You can open it by selecting the Bob icon beside the navigation bar. Bob understands natural language, allowing you to communicate with it as you would with a human developer, with no special commands required. Learn more about [the chat interface and its components](#).



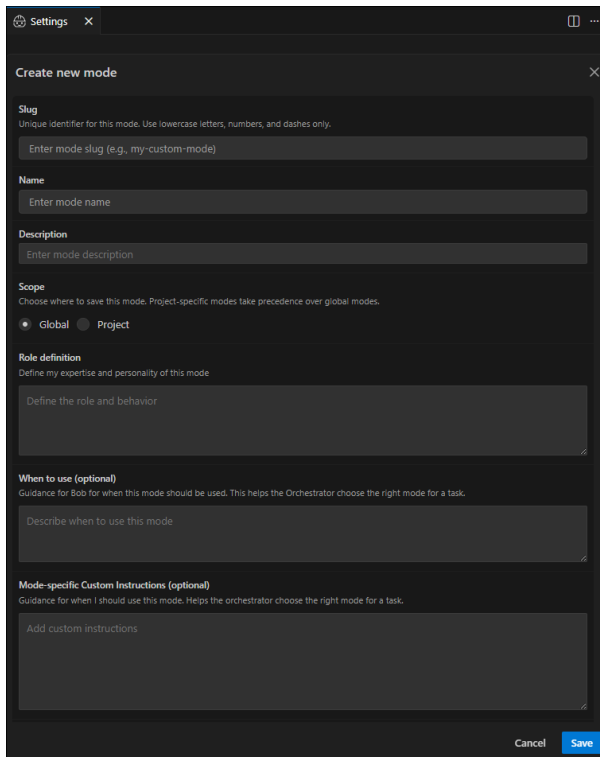
- [Modes](#)

Modes are specialized personas that tailor Bob's behavior for your specific tasks. Each mode offers different capabilities and access levels to help you accomplish particular goals more efficiently. Learn more about [modes and their usage](#).



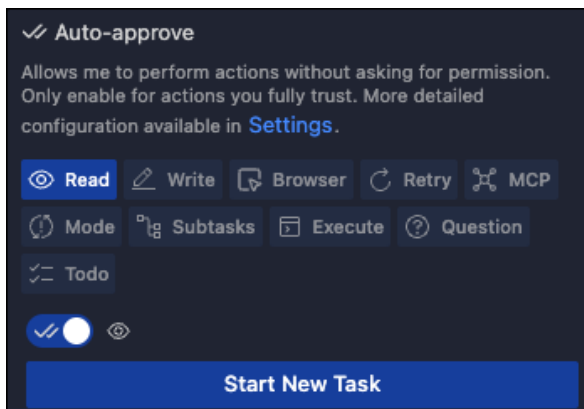
- [Custom modes](#)

Tailor Bob's behavior for specific tasks or workflows by creating custom modes. You can make modes global (available across all projects) or project-specific. Learn more about [custom modes and its configuration](#).



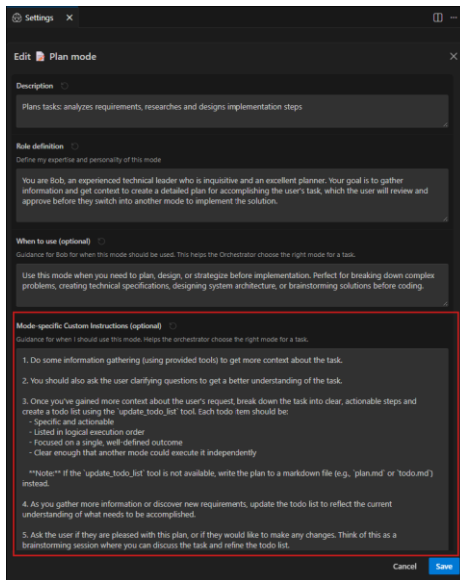
- [Auto-approve](#)

Auto-approve settings speed up your workflow by eliminating repetitive confirmation prompts, but they significantly increase security risks. Learn more about [auto-approve and its configuration](#).



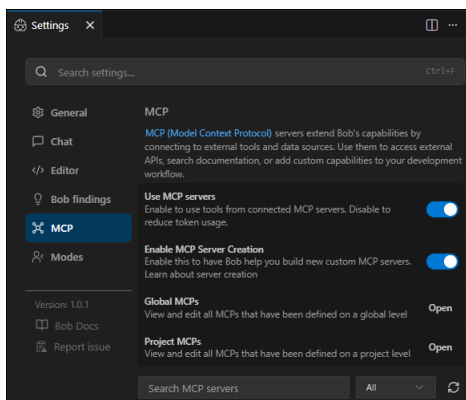
- [Custom instructions](#)

Custom rules influence how Bob responds to your requests, aligning output with your specific preferences and project requirements. You can control coding style, documentation approach, and decision-making processes. Learn more about [custom instructions and its configuration](#).



- [MCP servers](#)

The Model Context Protocol (MCP) extends Bob's capabilities by connecting to external tools and services. This guide shows you how to configure, manage, and use MCP servers with Bob. Learn more about [MCP server configuration with Bob](#).



- [Context mentions](#)

Context mentions let you reference specific elements of your project directly in your conversations with Bob. By using the @ symbol, you can point Bob to files, folders, problems, and other project components, enabling more accurate and efficient assistance. Learn more about [context mentions](#).

- [Security guidelines](#)

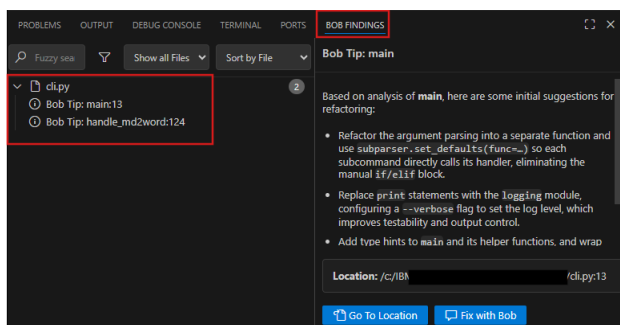
Bob has capabilities for coding and system interaction. To use safely, follow the [security guidelines](#).

```
> .bobignore

# Example .bobignore patterns
.env
secrets/
*.key
config/credentials.json
```

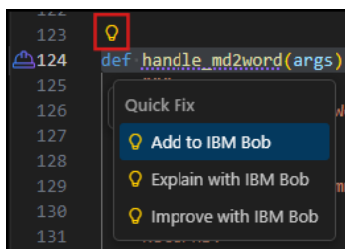
- [Bob tips](#)

Bob tips continuously monitors your code and identifies functions with quality issues in real-time. When functions are too complex, Bob marks them with purple underlines and provides refactoring suggestions. Learn more about [Bob tips and findings](#).



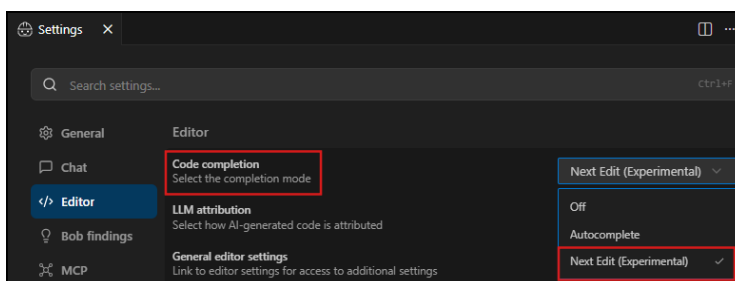
- [Code actions](#)

Code actions provide quick fixes, refactorings, and AI-powered suggestions directly in your editor. When you select code or encounter a problem, Bob offers contextual assistance through a lightbulb icon or context menu. Learn more about [utilizing code actions](#).



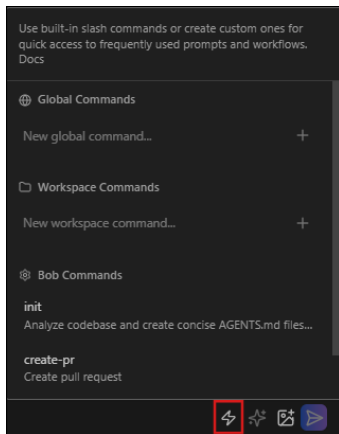
- [Next Edit](#)

Use Next Edit to get code completions and automatic error fixes directly in your editor. Learn more about [Next Edit and its usage](#).



- [Slash commands](#)

Create custom slash commands to automate repetitive tasks and extend Bob's functionality with simple markdown files. Learn more about [slash commands and its usage](#).



- [Checkpoints](#)

Checkpoints automatically version your workspace files during tasks, enabling non-destructive exploration of AI suggestions and easy recovery from unwanted changes. Learn more about [checkpoints and how work with it](#).

- [Code reviews](#)

Bob can review your code directly from your IDE, catching potential issues before you commit your work. Learn more about [code reviews and its usage](#).

Command	Use Case
<code>/review</code>	Quick code check
<code>/review --branch main</code>	Pre-PR review
<code>/review #123 --issue-coverage</code>	Verify issue completion
<code>/review --pr-desc</code>	Generate PR description
<code>/review file.py</code>	Focused file review

- [Commit messages](#)

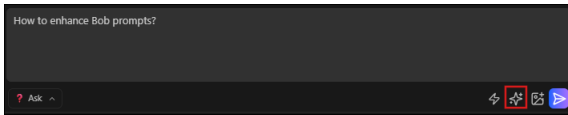
Bob can automatically generate meaningful commit messages based on your staged changes, saving you time and ensuring consistency in your commit history. Learn more about [commit message and its usage](#).

- [Pull requests](#)

Bob can generate pull requests (PRs) and PR descriptions directly from your IDE, streamlining your development workflow and saving time. Learn more about [pull requests and its usage](#).

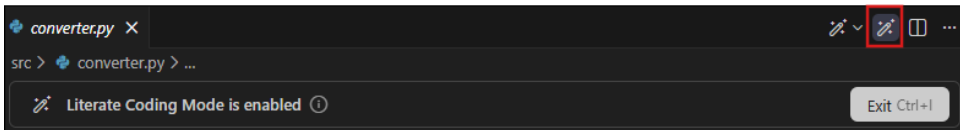
- [Enhance prompt](#)

Improve your prompts before sending them to Bob. Click the sparkles icon in the chat input to automatically refine your request, making it clearer, more specific, and more likely to yield better results. Learn more about [enhancing prompts](#).



- [Literate coding](#)

Literate coding allows you to write code with AI assistance directly inside your editor. Instead of switching to a chat window or writing long prompts, you type instructions in plain language right where the code should go. Bob then generates the implementation for you in context and shows a diff of the changes. Learn more about [literate coding and its usage](#).



- [Skills](#)

Reusable instruction sets that teach Bob new workflows and specialized tasks. Skills act as recipes that guide Bob through specific types of work in a consistent, repeatable manner. Learn more about [skills and its usage](#).

Bob IDE hands-on exercises

Try the quick start hands-on exercises for sample use cases to get started with using Bob.

Important note: You must use the [hackathon-provisioned IBM Bob account](#) to avoid usage on your personal Bob account, if you have one.

- [Bob exercises](#)

Quick hands-on exercises to learn how to use IBM Bob in the IDE for understanding code, refactoring, and automating everyday development tasks.

- [Bob and watsonx Orchestrate exercises](#)

Hands on exercises that show how IBM Bob and watsonx Orchestrate work together to design, build, and run agent-based workflows.

Bob exercises:

- [Quickstart exercise](#)

Learn to use IBM Bob for application modernization by upgrading a Node.js Express API from version 16 to 22. Try AI-assisted development with modes, approvals, and literate coding in this hands-on tutorial.

- [Travel demo app](#)

Get started with IBM Bob and the Galaxium Travels demo app by learning the tutorial flow, setup requirements, and application architecture.

- [Build agents with Bob /init](#)
Give Bob persistent project context across conversations and modes with /init, which automatically generates AGENTS.md files.
- [Get code predictions](#)
Accelerate development with Bob's tab completion and next edit prediction, which suggest context-aware code and automatically navigate you to likely follow-up edit locations.
- [Generate code from comments](#)
Use literate coding to have Bob generate code from natural language comments, where Bob helps you write precise, context-aware code modifications directly in your editor.
- [Plan and implement complex features](#)
Learn how to use the agentic chat sidebar to employ a Plan mode and Code mode workflow, where Bob plans complex features end-to-end and then implements them autonomously across your full codebase.
- [Standardize Bob's behavior](#)
Standardize Bob's behavior across your team using project-level rules files that tell Bob to document its code and remember its previous actions.
- [Add Bob capabilities](#)
Extend Bob's job capabilities by creating a custom product-manager mode with a tailored role definition, behavioral instructions, and deterministic tool access constraints.
- [Create a new context window](#)
Manage Bob's context window to preserve memory, control cost, and maintain output quality during complex or long-running conversations.

Bob and watsonx Orchestrate exercises:

- [Using IBM Bob to build watsonx Orchestrate agents and MCP tools](#)
A hands-on guide for automating the full process of building and deploying an agentic workflow using IBM Bob.
- [Build agentic workflows programmatically on watsonx Orchestrate using IBM Bob](#)
A hands-on guide for creating automated invoice-processing agentic workflows using IBM Bob to generate code, tools, and configuration for watsonx Orchestrate.
- [Enforce secure coding with AI: Eliminate hardcoded secrets using IBM Bob, Vault, and MCP Gateway](#)
Build a governed AI development workflow that prevents secret sprawl through automated scanning,

dynamic secret management, and enterprise-grade access control.

- [Turn BPMN diagrams into production-ready agents with Bob skills and watsonx Orchestrate](#)

A hands-on guide for using Bob skills to transform BPMN process models into SOP-driven, fully tested, and deployable watsonx Orchestrate agents with best-practice workflows, tools, and automation patterns.

2. Bob Shell (Optional)

Bob Shell brings IBM Bob's AI capabilities to your command line. Bob Shell delivers the same context awareness and reasoning-focused approach from IBM Bob but optimized for shell environments and automated processes.

Install Bob Shell

Follow the [Bob Shell installation instructions](#).

Authentication

When using Bob Shell, you will be prompted to authenticate with your internet browser and IBMid. After authenticating, close your browser and return to your Bob Shell instance.

Start an interactive session

Interactive sessions provide a conversational interface to Bob directly in your terminal, allowing real-time assistance with your development tasks. Learn more about [starting interactive sessions](#).

```

Welcome to
BOB
SHELL
Version 0.8.34

Here are some helpful commands to get started:

/clear      Clear the screen and conversation history
/bug       Submit a bug report. Optional title, /bug [title string].
up         Move to the previous completion option.
return     Submit the current prompt.
/settings  View and edit Bob-Shell settings

Sandbox mode Disabled

You are running Bob-Shell in your home directory. It is recommended to run in a project-specific directory.

Enter your prompt, / for commands, @ for files, ! for Shell mode
Auto-approve: Off  Tokens left: 180x  19.32/5000 (99%) | Current: 0.00  (F12 for details)  Mode: Code

```

Start a non-interactive session

Non-interactive session provides a method to use Bob Shell directly from the command line without entering an interactive session. Use for automation, scripting, and batch processing tasks. Learn more about [starting non-interactive sessions](#).

Bob Shell configuration

You can configure Bob Shell to match your workflow preferences. Learn more about [Bob Shell configuration](#).

Bob Shell usage examples

Try practical examples on how to use Bob Shell for debugging, code improvement, file creation, documentation generation, and learning new concepts. Learn more about [Bob Shell usage examples](#).

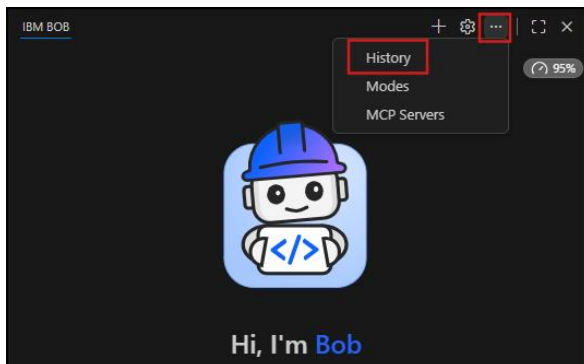
Exporting Bob task session reports for judging

For judging purposes, **each team member** must export and upload all relevant Bob task session reports as part of your project submission. These reports must be included in your **final code repository** as **submission deliverable**.

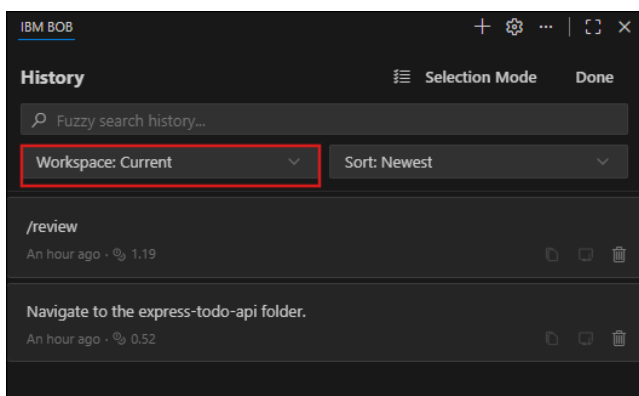
Important: Ensure that all credentials and API keys are removed in your code before exporting the task session reports and uploading them to your public code repository which will be shared as part of your submission. If any IBM Bob or Cloud service credentials are detected in your code repository, the IBM Security team will deactivate your account access.

Follow the steps below to complete this process:

- Create a folder named **bob_sessions** in your project submission code repository.
- In your Bob IDE's chat interface, select the **Views and More Actions** and then select the **History** option. The History tab will appear.



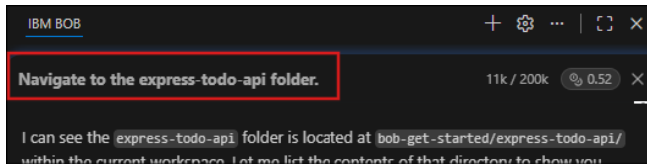
- In the history view, confirm that you are in the correct project workspace and that it is current. If your submission includes tasks from multiple workspaces, select **All** to view tasks across all relevant workspaces.



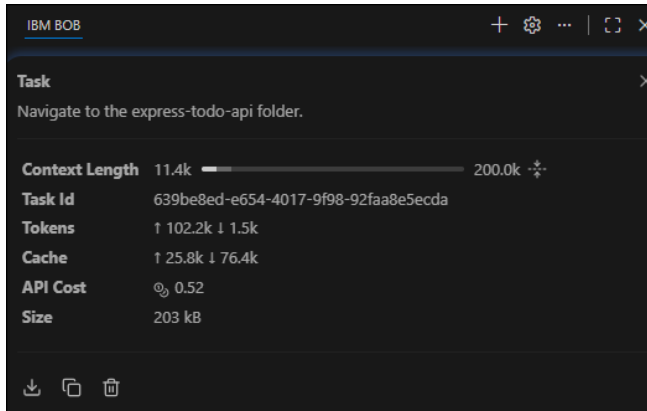
- In the task history list, select a task related to your project submission. The selected task will open in

the chat panel.

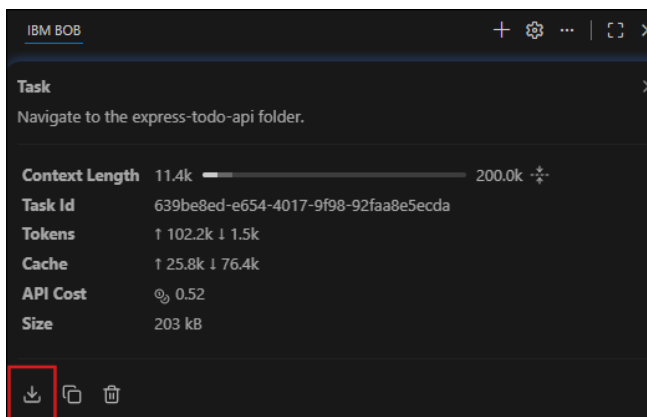
- Select the task header. A task session consumption summary will be displayed.



- Take a screenshot of the task session consumption summary.



- From the same task session consumption summary view, select the **Export task history** icon to download the task history as a markdown file.



- Now, repeat the steps above for all the tasks related to your project submission.
- Upload all task session consumption summary screenshots and exported task history markdown files into the **bob_sessions** folder in your code repository.

Important: Before submitting your project, ensure that your team final code repository includes the **bob_sessions** folder with all required screenshots and exported task history reports relevant to the project solution.

Get started with IBM watsonx (Optional)

To access and use IBM watsonx products for this hackathon, participants must be registered for the hackathon, and request for a pre-configured IBM Cloud account. This account will provide the necessary environment to work with watsonx.ai, watsonx Orchestrate and other supporting products for this hackathon.

Note on teaming

If you are working as part of a team, please plan to collaborate using one team member's cloud account. Adding or removing team members is not supported on the hackathon-provisioned IBM Cloud accounts.

Note on available services

The IBM Cloud account provisioned for this hackathon is **pre-configured with only the services required** to complete the hackathon. You will not be able to configure any new service or modify permissions for existing ones due to restricted access. If you notice a permission/access issue for any service or the cloud catalog, then they are not required/available for this hackathon.

The following optional IBM Cloud services are provided for your hackathon solution building:

- [Natural Language Understanding](#)
- [Speech-to-Text](#)
- [Text-to-Speech](#)
- [Cloudant](#)

Note on preventing exposure of IBM Cloud credentials

Hackathon participants may use IBM Cloud credentials to build their solutions that leverage IBM technologies on IBM Cloud. During development, testing, collaboration, or while submitting their final project in a public repository, participants may unintentionally expose these credentials on publicly accessible platforms.

Exposing IBM Cloud API keys or any cloud credentials can lead to unauthorized access, misuse of resources, and account suspension. If any IBM Cloud credential associated with your hackathon account is detected in a public repository or publicly accessible platform:

- The credential will be deactivated immediately.
- Your hackathon cloud account access will be suspended until you:
 1. Remove the exposed credential from all public sources. If you are using GitHub, you can refer to [Removing sensitive data from a repository](#).
 2. Rotate and replace the exposed credential.
 3. After completing these steps, confirm remediation with the hackathon support team and request access to your hackathon cloud account.

Please follow the [Preventing exposure of IBM Cloud credentials in public repositories and platforms guide](#) for best practices to avoid disruptions to your project and ensure the security of your IBM Cloud resources.

Note on IBM Cloud service usage

For this hackathon, **\$80 credits** will be automatically applied to your provisioned **IBM Cloud account**. These credits should be sufficient for designing and creating compelling submissions using the services available on your account.

You will receive periodic email notifications about your **credit consumption** at the following usage levels: **25%**, **50%**, and **80%**. Once you reach **100% usage**, your account will be **suspended**. You can appeal the suspension by completing the form shared in the account suspension notification email.

Please note that these email notifications are sent **once per hour**, so there is a possibility that you may **exhaust all your credits before receiving an alert**.

We recommend planning your usage carefully and backing up your work regularly to avoid disruptions.

Request your hackathon IBM Cloud account

Follow the steps below to request your cloud account:

1. Access the hackathon’s IBM Cloud account request URL in your browser: <https://www.ibm.com/account/reg/us-en/signup?formid=urx-54370>.
2. Create an IBMid if you have not already created one using the same email you used to register for the hackathon by completing the account information. Then click the “**Next**” button.

If you already have an IBMid that uses the same email you used to register for the hackathon, proceed to **log in**, complete the authentication process and skip to **step 4**.

The screenshot shows the 'Create an IBMid' form for the IBM Bob Hackathon. On the left, there is a sidebar with the IBM logo and the text 'IBM Bob Hackathon'. Below this, it states: 'Now that the hackathon has started, you can request an IBM Cloud account to access and use the IBM watsonx Orchestrate platform. This account is only to use during and for the purposes of this hackathon and will be closed on **May 19, 2026 at 10 AM ET**. Once you have submitted an account request, please allow 1 hour for your account to become active. Check the email address seen in this form, and check your junk/spam folder if you are not able to find the email in your inbox. **Start date:** May 15, 2026 at 10 AM ET **Submissions due:** May 17, 2026 at 10 AM ET'. The main form area is titled 'Create an IBMid' and has a link 'Already have an IBM account? Log in'. There is a 'Sign up with Google' button. Below that is a section titled 'Account information' with a dropdown arrow. It contains the following fields: 'Business email' (gju@...@ibm.com), 'Password' (masked with dots), 'First name' (John), 'Last name' (Doe), and 'Country or region of residence' (Canada). At the bottom of the form is a 'Next' button.

3. Verify your email by entering the 7-digit code sent to your email and select the “**Submit**” button.

Create an IBMid

Already have an IBM account? [Log in](#)

Account information

Verify email

We emailed a 7 digit code to gao[redacted]@gmail.com
This code will expire in 30 minutes.

Verification token
3526108

Didn't receive the email? Check your spam filter for an email from ibmacct@iam.ibm.com.

You can resend the verification token in 19 seconds

I'd like IBM to use my contact details to keep me informed about products, services, and offers. More information on how IBM uses data and ways to [opt-out](#) can be found in the [IBM Privacy Statement](#).

I accept the product [Terms and Conditions](#).

Submit

4. Complete the IBM Cloud account request by verifying your name and email and click the “**Request account**” button.

Note: If you see the message “Submission error Unable to verify registration” it means you are not registered for the hackathon and cannot request a cloud account.

IBM

IBM Bob Hackathon

Now that the hackathon has started, you can request an IBM Cloud account to access and use the optional IBM watsonx.ai and watsonx Orchestrate platforms. This account is only to use during and for the purposes of this hackathon and will be closed on **May 19, 2026 at 10 AM ET**.

Once you have submitted an account request, please allow 1 hour for your account to become active. Check the email address seen in this form, and check your junk/spam folder if you are not able to find the email in your inbox.

Start date: May 15, 2026 at 10 AM ET
Submissions due: May 17, 2026 at 10 AM ET

Complete the request

Name
Go [redacted]

Email
gao[redacted]@gmail.com

Access code (optional)

Optional: If you were provided an access code by the event organizer, enter it here

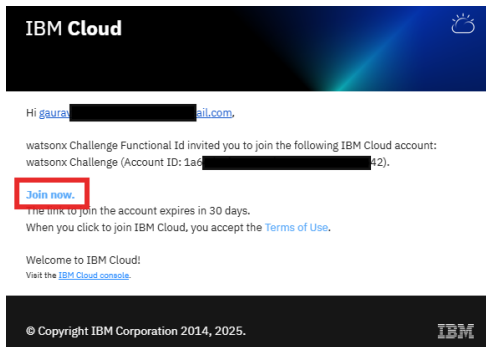
Request account →

5. Once your request is successfully submitted, you will receive an email invite to join the cloud account. Next, follow the [Accessing your hackathon IBM Cloud account](#) instructions to access your account.

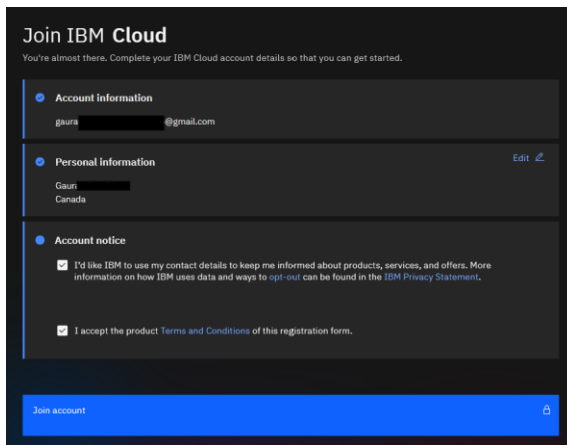
Accessing your hackathon IBM Cloud account

Once your team has been provisioned an IBM Cloud account, all team members will receive an email invite to join the hackathon-provisioned cloud account. Follow the steps below to access your team’s cloud account:

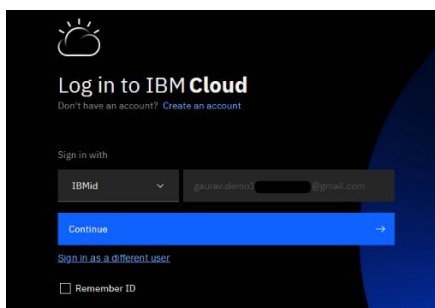
4. Check the email inbox you used to register for the hackathon and open the email you received from the IBM Cloud team about joining your cloud account. Please check your junk/spam folders if you are not able to find the email in your inbox. You can also quickly search for “IBM Cloud” to locate the email.
5. Click the **Join Now** button seen in that email. A new browser tab will open with the cloud account sign up page.



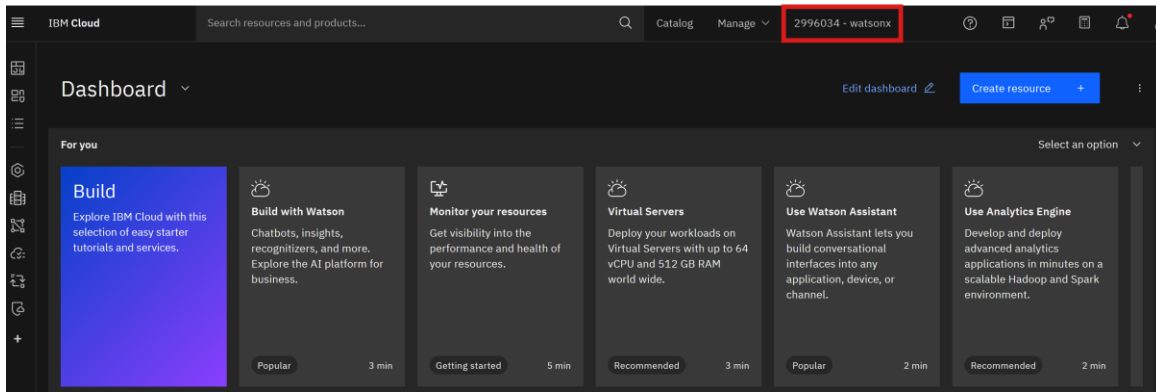
- Review your account and personal information. Read and accept the Account notice and click the **Join Account** button.



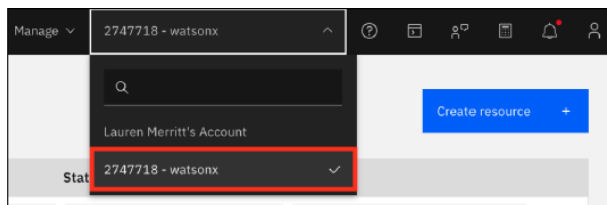
- Complete the authentication process by clicking the Continue button.



- After you authenticate successfully, you will be taken to the IBM Cloud dashboard.



- If you have an existing personal IBM Cloud account for the same email/IBMid, sometimes you will be directed to your personal account. In this case, please switch your account to the **xxxxxxx - watsonx** account. Select your account drop-down at top-right of the dashboard and select watsonx account. Refer to the below image on switching accounts in your cloud dashboard.



Accessing and utilizing IBM watsonx products

To begin building your solution, explore the capabilities and resources for each IBM watsonx product enabled for this hackathon.

- [IBM watsonx Orchestrate \(Optional\)](#)

IBM watsonx Orchestrate is an intuitive, AI-powered platform that you can use to create, configure, and deploy intelligent agents that can automate business tasks. Whether you're automating repetitive workflows or building complex multi-agent systems, the platform is designed to support users of all skill levels.

- [IBM watsonx.ai \(Optional\)](#)

A powerful AI studio that enables you to experiment with foundation models like IBM Granite through the Prompt Lab. It can be used to enhance your agents with natural language understanding, intelligent decision-making, and dynamic response generation—serving as an inference engine to make your AI agents smarter and more interactive.

1. IBM watsonx Orchestrate

After successfully [joining the IBM Cloud account](#), you can access watsonx Orchestrate to begin working on the platform and building your solution.

With watsonx Orchestrate, you can:

- Build and deploy agents
- Automate tasks such as scheduling, data entry, and approvals
- Start small and scale up with advanced tools and integrations
- Explore the clean, guided UI that simplifies agent creation and management.

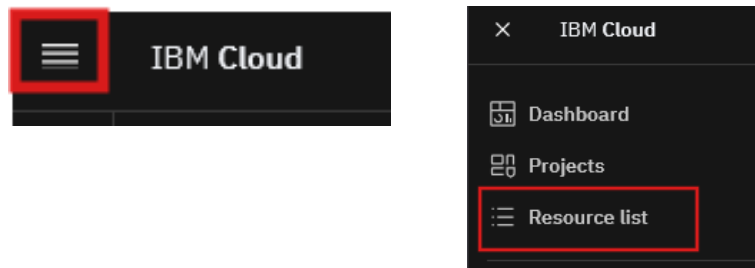
Explore a few [demos](#) on how to use watsonx Orchestrate.

These watsonx Orchestrate features/capabilities are out of scope for this hackathon:

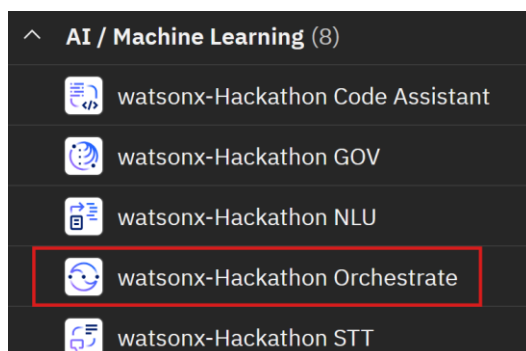
- Build with AI (Preview)

Accessing watsonx Orchestrate

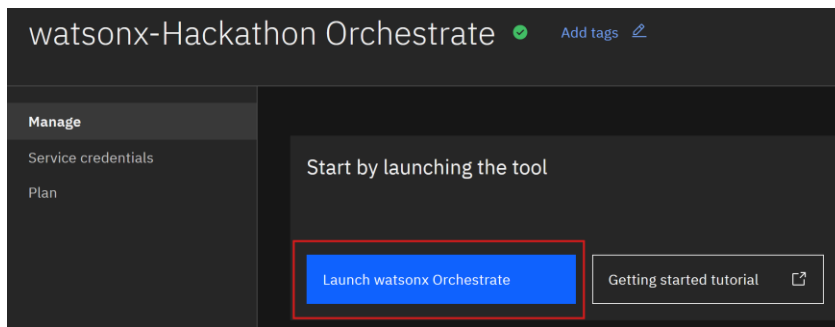
6. In your IBM Cloud account dashboard, select the **Navigation menu** on the top left of the dashboard and select the **Resource list** option.



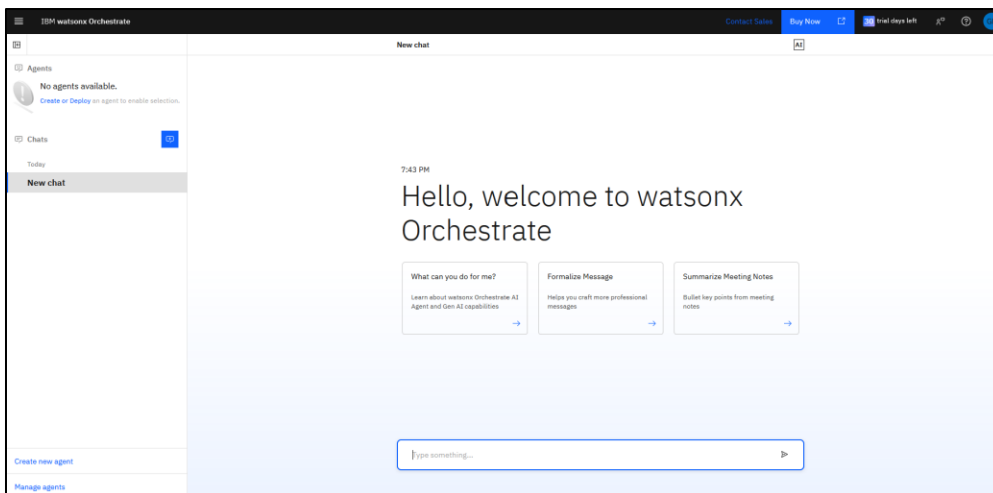
7. Expand the **AI / Machine Learning** section and select **watsonx-Hackathon Orchestrate** service.



- You will be navigated to **watsonx-Hackathon Orchestrate** service instance dashboard. Select the **Launch watsonx Orchestrate** button.

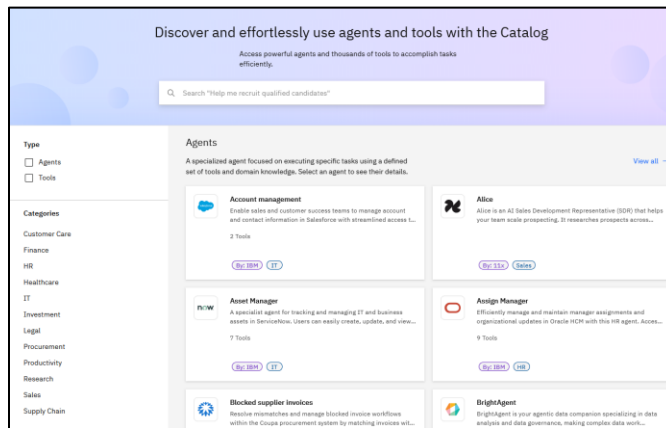


- You will be navigated to the watsonx Orchestrate platform with a welcome message and a new chat window.



Discover the catalog

The IBM watsonx Orchestrate catalog is your gateway to a rich collection of prebuilt AI agents and tools, designed to support a wide range of business functions and use cases. Whether you're looking to automate tasks, enhance productivity, or integrate with backend systems, the catalog helps you find the right solutions quickly and efficiently. [Learn more about discovering the catalog.](#)



Building agents

In IBM watsonx Orchestrate, agents are a key component of the agentic AI framework, enabling you to create complex, dynamic systems that can adapt and respond to changing conditions.

By building agents, you can:

- Automate repetitive tasks
- Improve decision-making processes
- Enhance customer and employee experiences
- Increase operational efficiency

1. Prepare to build AI agents

Developing intelligent, scalable, and reliable agents in watsonx Orchestrate requires a strategic, multi-phase approach. It goes beyond writing prompts or connecting APIs. Successful agents need thoughtful planning, structured development, testing, and ongoing governance. The following overview breaks down each phase of agent development, highlighting key considerations. Learn more about [preparing to build AI agents](#).

2. Creating and customizing agents

Creating and customizing an AI agent involves defining its purpose and personality through thoughtful descriptions and selecting the most suitable foundation model that aligns with your brand or use case. You can also configure key elements like the welcome message and starter prompts to ensure the agent engages users effectively from the first interaction. Together, the configured components shape how the agent communicates, responds, and delivers value across conversations.

Learn [how to create an agent from scratch or from a template](#).

Create an agent

Create from scratch **Create from template**

Build your custom agent step by step to create a custom solution.

Browse the catalog and use the attributes of another agent as a template to create your agent.

Name

Name your agent

Description

Example: This agent helps answer customer questions about return and cancellation policies.

Describe your agent, including its purpose, tools, and agents, to help other agents and users know when to use it.

[What makes a good description?](#)

Cancel Create

Expanding your agent's capabilities:

To enhance your agent's abilities, explore the following configuration options:

- **Define a profile**

Provide a clear and specific description of the agent's purpose. This helps in multi-agent orchestration by enabling accurate selection based on capabilities. See [Defining the description of your agent](#).

Agent chat / Manage agents / agent-1

agent-1 AI Model: llama-3-2-90b-visi...

Profile

Describe what your agent does so people and other agents know when to use it.

Description*

This agent helps with basic queries and can respond to any requests that leverage the tools integrated into it.

Describe your agent, including its purpose, tools, and agents, to help other agents and users know when to use it.

[What makes a good description?](#)

Welcome message

Customize the welcome message from the agent on the home screen.

Welcome message 37/100

Hello, welcome to watsonx Orchestrate

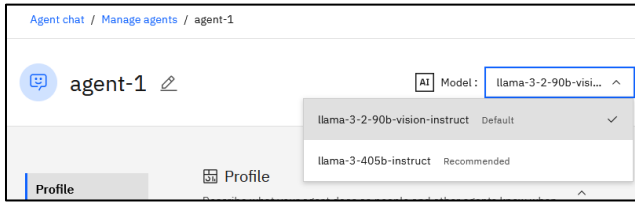
Quick start prompts

Pre-set messages sent to the agent to start the conversation.

[Add prompt](#) +

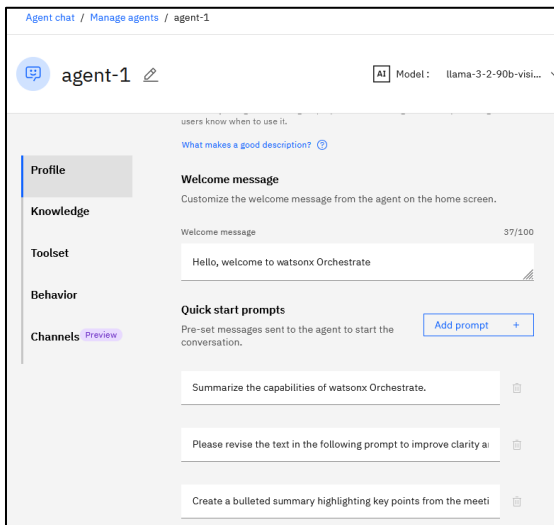
- **Choose an AI model**

Choosing the right AI model helps your agent understand user intent, reason through tasks, and generate reliable responses. Your choice directly affects accuracy, performance, and cost. See [Choosing an AI model](#).



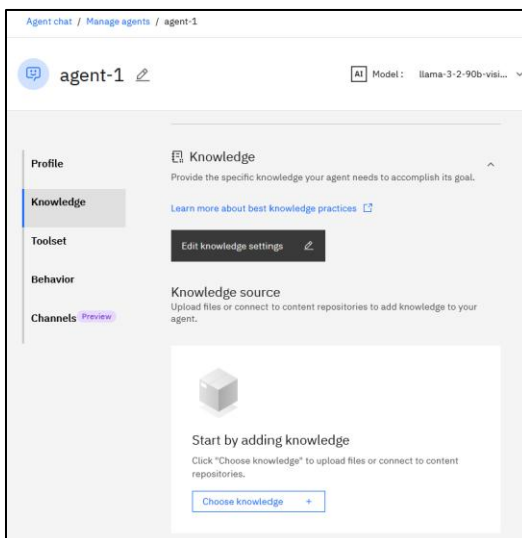
- **Customize the welcome message and starter prompts**

Customize the welcome message and starter prompts to guide users when they begin interacting with the agent. See [Customizing the welcome message and starter prompts](#).



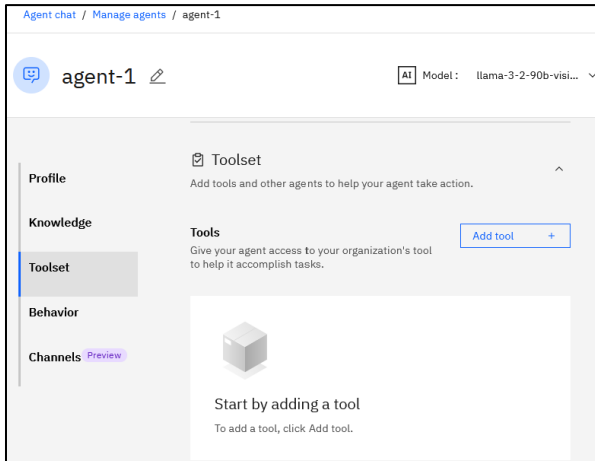
- **Add knowledge**

Enhance the agent's domain expertise by adding contextual knowledge from files or content repositories. See [Adding knowledge to agents](#).



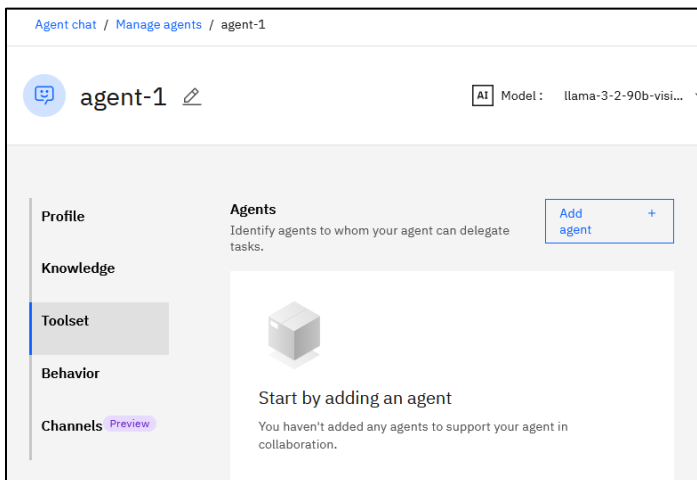
- **Add and manage tools**

Integrate tools to enable the agent to perform automated tasks such as retrieving data or sending emails. See [Adding and managing tools for agents](#).



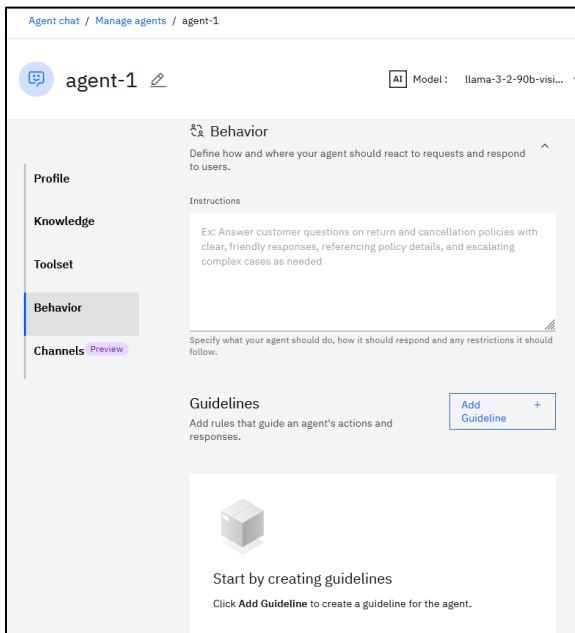
- **Orchestrate agents**

Enable multi-agent orchestration by adding collaborator agents that work together to achieve shared goals. See [Orchestrating agents](#).



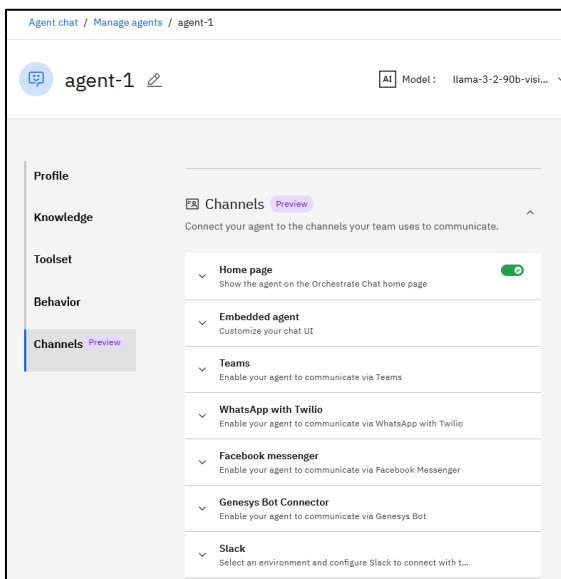
- **Add instructions**

AI agents operate based on a set of behavioral instructions that define how they respond to user inputs, perform tasks, and interact across different channels. These instructions serve as the agent's internal guidelines, helping it act consistently and effectively in various scenarios. See [Adding instructions to agents](#).



- **Connect agents to channels**

Channels are the interfaces through which you can interact with a conversational agent, such as messaging platforms and voice assistants. Integrate your agent with these channels to access services and support directly within your preferred environments to enable seamless and scalable communication. See [Connecting agents to channels](#).

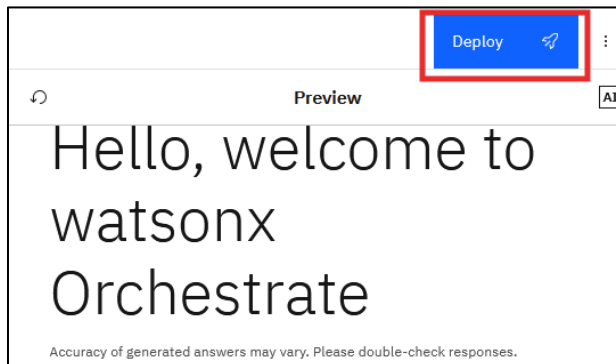


- **Configure rich responses**

Enable rich responses to incorporate multimedia and structured elements, making AI interactions clearer, more efficient, and engaging for users. See [Configuring rich responses from the AI assistant builder](#).

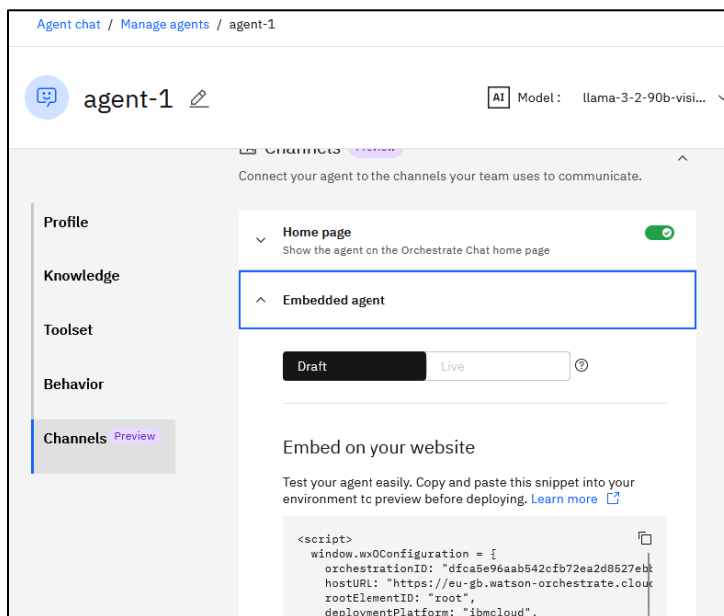
- **Deploy agents**

Finalize the setup by deploying the agent to make it available in live environments such as chat interfaces. See [Deploying agents](#).



- **Embed agents in applications**

IBM watsonx Orchestrate makes it easy to embed intelligent agents directly into your web applications by using the embedded chat feature. It enables interactive, contextual conversations while maintaining enterprise-grade security and flexibility. See [Embedding agents in applications](#).



- **Building agents using the ADK**

You can build powerful, customizable agents using the IBM watsonx Orchestrate Agent Development Kit (ADK). [Learn more about using the ADK](#).

3. Building tools

Tools act as interfaces to external capabilities, enabling agents to interact with systems, retrieve data, run

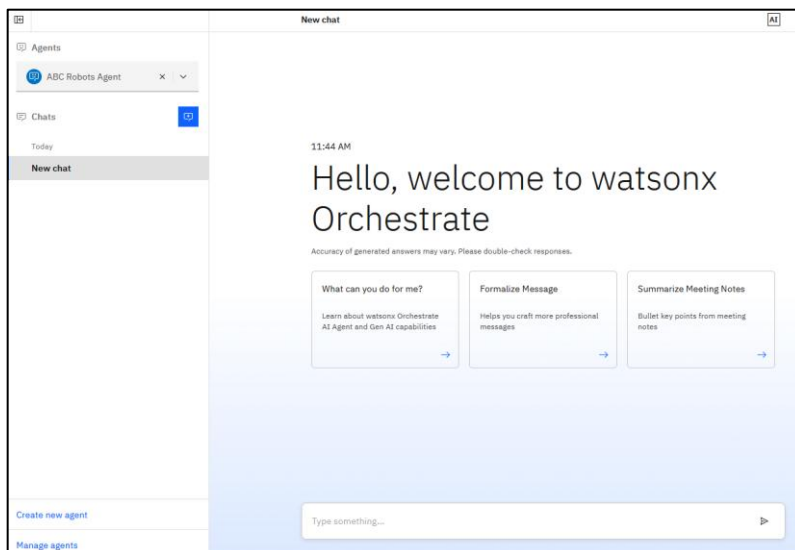
calculations, or trigger workflows that go beyond their built-in reasoning. You can build new tools and edit existing tools for your agents to use.

Following are the different ways to add a tool:

- [Creating agentic workflows](#)
- [Adding tools to an agent](#)
 - [Catalog](#)
 - [Local instance](#)
- [Importing external tools](#)
 - [MCP server](#)
 - [OpenAPI](#)

Using agents in the chat

In IBM watsonx Orchestrate, agents collaborate to automate tasks and manage workflows. [Learn more using agents in Orchestrate Chat.](#)



Managing app connections

To use the external applications within IBM watsonx Orchestrate, you must establish a connection between them which acts as bridge enabling communication between watsonx Orchestrate and the external applications. [Learn more about managing app connections and credentials.](#)

watsonx Orchestrate API

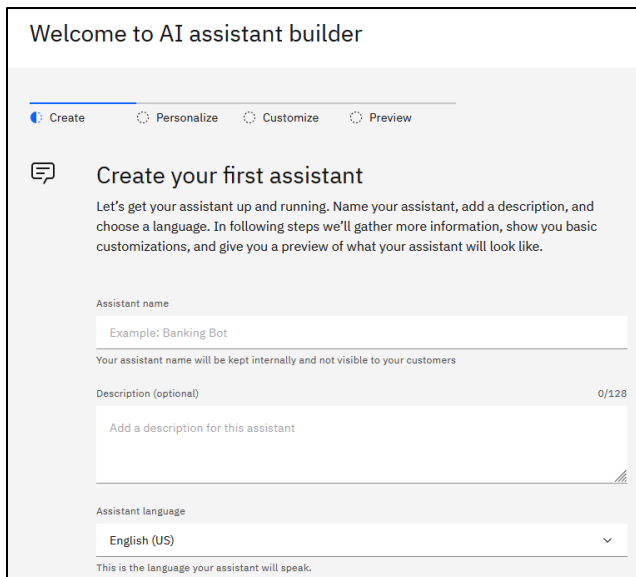
IBM watsonx Orchestrate provides a set of APIs to boost your experience using the features from the product.

- [Getting started with the API](#)
Begin using the API with practical, step-by-step guidance.
- [API reference](#)
Get the methods that you can use to call custom projects, or other resources from the AI chat.

Building AI assistants

In IBM watsonx Orchestrate, you build the AI assistant by using AI assistant builder. AI assistant builder is a chat interface builder that helps to deploy an engaging and embedded chatbot experience. AI assistant builder integrates the power of large language models (LLMs) and conversational capabilities of watsonx Assistant to enable responsive and interactive conversation between the users and watsonx Orchestrate.

To learn more, see [Building AI assistants in AI assistant builder](#).



The screenshot displays the 'Welcome to AI assistant builder' interface. At the top, there are four navigation tabs: 'Create' (active), 'Personalize', 'Customize', and 'Preview'. Below the tabs, the main heading is 'Create your first assistant'. A sub-heading reads: 'Let's get your assistant up and running. Name your assistant, add a description, and choose a language. In following steps we'll gather more information, show you basic customizations, and give you a preview of what your assistant will look like.' The form includes three main sections: 1. 'Assistant name' with a text input field containing 'Example: Banking Bot' and a note: 'Your assistant name will be kept internally and not visible to your customers'. 2. 'Description (optional)' with a text area containing 'Add a description for this assistant' and a character count '0/128'. 3. 'Assistant language' with a dropdown menu set to 'English (US)' and a note: 'This is the language your assistant will speak.'

Quick start hands-on exercises

Try the quick start hands-on exercises for sample use cases to get started with using watsonx Orchestrate:

Important notes:

- You must use the hackathon-provisioned IBM Cloud account to access and use watsonx Orchestrate platform to try the below sample exercises.

watsonx Orchestrate SaaS sample exercises:

- [Develop agents with no code using watsonx Orchestrate](#)
A hands-on guide for developing a multi-agent system in watsonx Orchestrate with no code that can communicate with the agents on various channels.
- [Creating intelligent, reusable agentic workflows on watsonx Orchestrate with no code](#)
A hands-on guide for creating AI-driven, human-in-the-loop workflows without writing code.
- [Build context-aware AI agents with watsonx Orchestrate and Astra DB](#)
AI agents are only as effective as their data, making precise retrieval essential for reliable results.
- [AgentOps in watsonx Orchestrate: Observability for Agents with Langfuse and IBM Telemetry](#)
A developer's guide for instrumenting and observing Agents with Langfuse and IBM Telemetry in watsonx Orchestrate.
- [Building and deploying prebuilt domain agents in watsonx Orchestrate](#)
Explore how to work with the prebuilt domain agents and integrate them into existing workflows.
- [Build a scheduled agentic workflow for daily notifications](#)
A practical example of building, importing, and scheduling an agentic workflow that automatically sends daily email notifications using watsonx Orchestrate.
- [Multi-agent orchestration with watsonx Orchestrate](#)
A hands-on journey to design, extend, and integrate AI agents by using Langflow, IBM Granite models, Astra DB, the AI Gateway, and a custom user interface on watsonx Orchestrate.

watsonx Orchestrate Agent Development Kit (ADK) sample exercises:

- [Getting Started with watsonx Orchestrate Agent Development Kit](#)
IBM watsonx Orchestrate includes the Agent Development Kit (ADK) which is a set of developer-focused tools for building, testing, and managing agents. With the ADK, developers get the freedom and control to design powerful agents using a lightweight framework and a simple CLI. You can define agents in clear YAML or JSON files, create custom Python tools, and manage the entire agent lifecycle with just a few commands.
- [Build an AI agent with Langflow, Granite 4.0 models, and watsonx Orchestrate](#)
A hands-on guide for creating Langflow tools, consuming them with watsonx Orchestrate Agent Development Kit, and using IBM Granite 4.0 Micro model for inference.
- [Connecting to MCP tools with watsonx Orchestrate](#)

A hands-on guide for creating MCP tools and consuming them with watsonx Orchestrate Agent Development Kit.

- [Extend Your AI Agents with External LLMs Using watsonx Orchestrate and AI Gateway](#)

A hands-on guide to bring your own model (BYOM) into watsonx Orchestrate using AI Gateway.

- [Create consistent evaluation workflows for AI agents](#)

A hands-on guide to setup and use a structured evaluation strategy to test, benchmark, and improve AI agent performance.

2. IBM watsonx.ai

After successfully [joining the IBM Cloud account](#), you can access watsonx.ai to begin working on the platform and building your solution.

Note on IBM watsonx.ai service usage

For this hackathon, **\$80 credits** will be automatically applied for the IBM services provisioned on the cloud account. This should be sufficient for designing and creating compelling submissions.

You will receive periodic email notifications about your **credit consumption** at the following usage levels: **25%**, **50%**, and **80%**. Once you reach **100% usage**, your account will be **suspended**. You can appeal the suspension by completing the form shared in the account suspension notification email.

Please note that these email notifications are sent **once per hour**, so there is a possibility that you may **exhaust all your credits before receiving an alert**.

IBM watsonx.ai service consumes the credits when used. Please plan to use it efficiently and back up your work accordingly. Refer [tips to work efficiently on watsonx.ai platform \(Tokens and CUH explained\)](#) and [saving your work](#).

Important:

- **Foundation model inferencing** consumes tokens, which are measured as Resource Units (RUs). **1,000 tokens = 1 RU**, and each RU costs **\$0.0001 USD**.
[Learn more about tokens and tokenization](#).
- If you are using **Jupyter Notebook editor on watsonx.ai**, consider selecting a **lower runtime environment** to avoid high resource consumption and quickly depleting your credits. Notebook runtimes are billed based on **Capacity Unit Hours (CUH)** at a rate of **\$1.02 USD per CUH**.
[Learn more about capacity unit hours and watsonx.ai Studio pricing plans](#).

Note on available watsonx.ai capabilities

The watsonx.ai platform is **pre-configured with only the services required** to complete the hackathon. If you notice a permission/access issue for any service or the cloud catalog, then they are not required/available for this hackathon.

These features/capabilities are out of scope for this hackathon:

- Agent Lab (Beta)
- Bring your own model
- Fine tuning models
- AutoAI pipeline
- AI governance
- Evaluation Studio
- SPSS Modeler

DO NOT USE the below listed watsonx.ai models as they are out of scope for the hackathon and can negatively impact the judgment of your project submission.

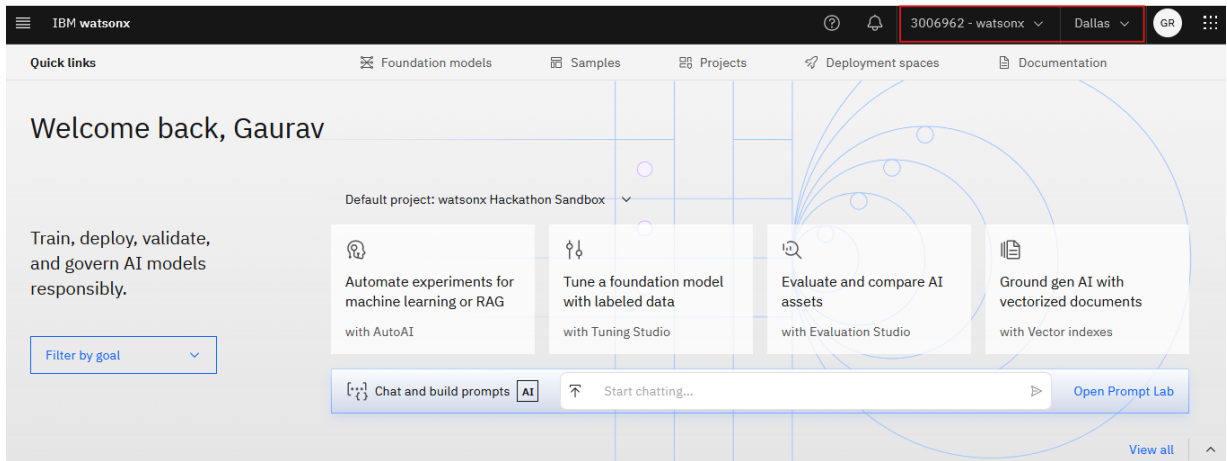
- llama-3-405b-instruct
- mistral-medium-2505
- mistral-small-3-1-24b-instruct-2503

The hackathon-provisioned IBM Cloud account will be deactivated after the completion of the hackathon. Please plan to [save your work](#) at the end of the hackathon.

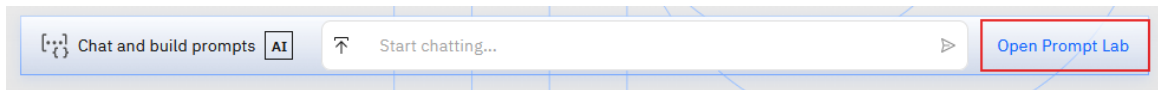
Access Prompt Lab on watsonx.ai

After successfully joining the IBM Cloud account, you can now access the Prompt Lab on watsonx.ai platform to work with the AI models supported on the platform and build your solution.

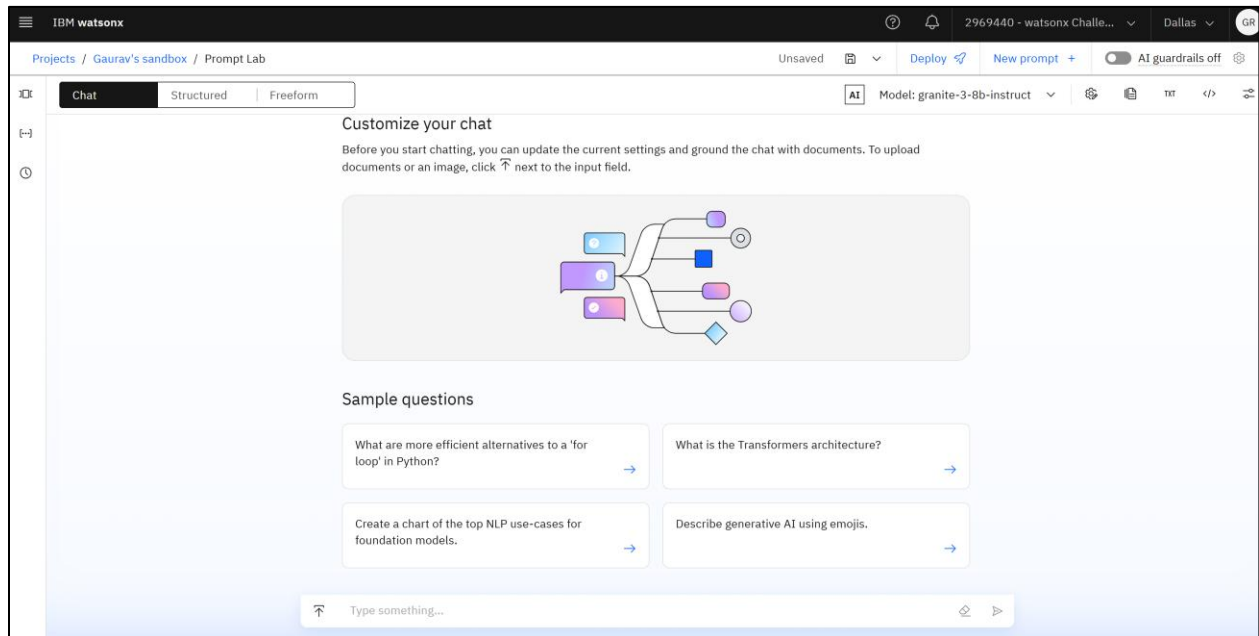
1. Log in to the [watsonx.ai platform](#) with the email you used to access your IBM Cloud account.
2. After successful authentication, you will see “Welcome to watsonx” widget. You can either take the tour or skip it.
3. Next, you will see the watsonx.ai dashboard. Ensure the name of the account is “**xxxxxxx – watsonx**” and the region is “**Dallas**”.



4. Select the **Open Prompt Lab** button on the “Chat and build prompts with foundational models” widget.



5. The “Welcome to Prompt Lab” tour will be displayed. You can take the tour to get a quick introduction or skip it.
6. The Prompt Lab Editor opens with a chat window to get you started with the prompt session.



Work with the watsonx.ai Prompt Lab

The watsonx.ai Prompt Lab is an easy-to-use prompt engineering interface where you can experiment with prompting different AI foundation models, explore sample prompts, tune model parameters, integrate

applications with an API endpoint, and save and share your best prompts.

[Take a tour of the Prompt Lab](#) and try the [interactive demo](#).

You can access and use the AI models to build your innovative solution using Prompt Lab.

Prompt Lab editor

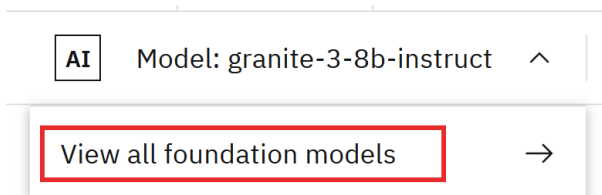
In the Prompt Lab, you can experiment with prompting different foundation models, explore sample prompts, as well as save and share your best prompts. The Prompt Lab editor is a great place to experiment and iterate with your prompts. Try the [quick start lab](#).

However, you can also prompt foundation models in watsonx.ai programmatically. Refer to “[Programmatic access \(API/SDK\)](#)” section.

Selecting an AI model

A default AI model will be pre-selected in the Prompt Lab editor. You can either use the same model or change to a different model. To select a different model:

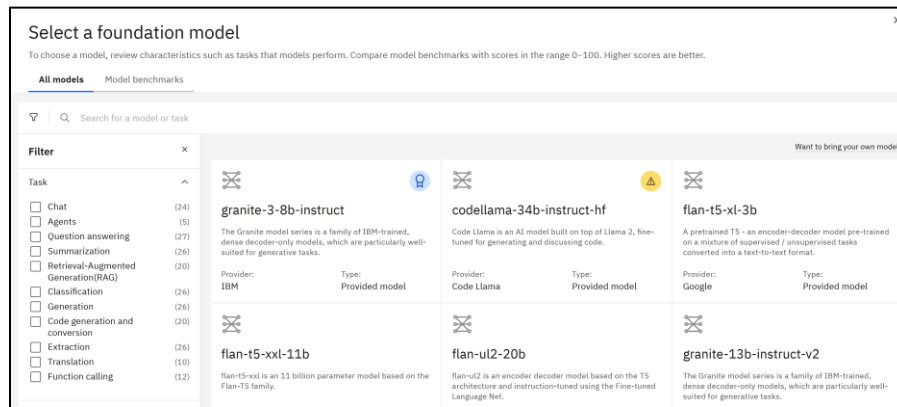
1. Select the AI Model drop-down menu at the top-right of the editor and select **View all foundation models**.



2. A foundation model selection widget appears. Clear the filters to see all the available models. You can use the filters to choose the right model for your solution building. You can select a model tile to learn about the model and use it. If you are limited to only “Chat” supported models, change the Prompt Lab editor to [Structured](#) or [Freeform](#) view and try selecting the models to see all the available model options.

Important: DO NOT USE the below listed watsonx.ai models as they are out of scope for the hackathon and can negatively impact the judgment of your project submission.

- llama-3-405b-instruct
- mistral-medium-2502
- mistral-small-3-1-24b-instruct-2503



To understand how models can address your use case, including information on model modalities, supported languages, tuning, and indemnification, see our product documentation on [choosing a model](#).

Note: Bigger models are not always better. [Learn](#) why smaller models can be better and more cost effective.

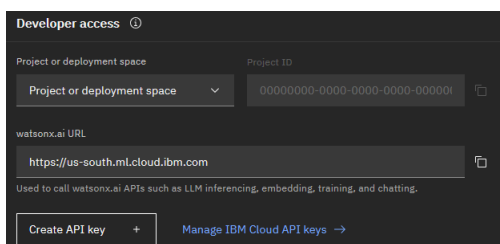
Programmatic access (API/SDK)

You can inference the watsonx.ai models with API or SDK requests.

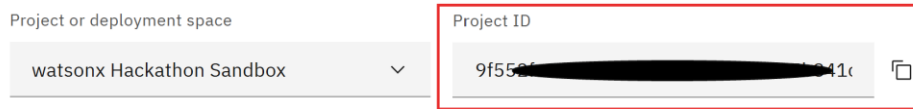
Developer access information

To use the supported watsonx.ai APIs/SDKs, you will need three values: a **project ID**, an **endpoint URL** and an **API key**.

- Go to [watsonx.ai home page](#).
- Scroll down to the “**Developer access**” section.



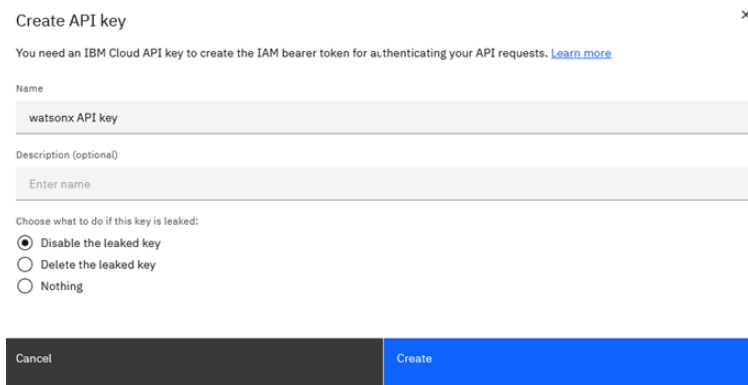
- Select the **Project or deployment space** drop-down and choose the **watsonx Hackathon Sandbox** option if you are working with that project, or select any other listed project that you are working with. A **project ID** will be displayed.



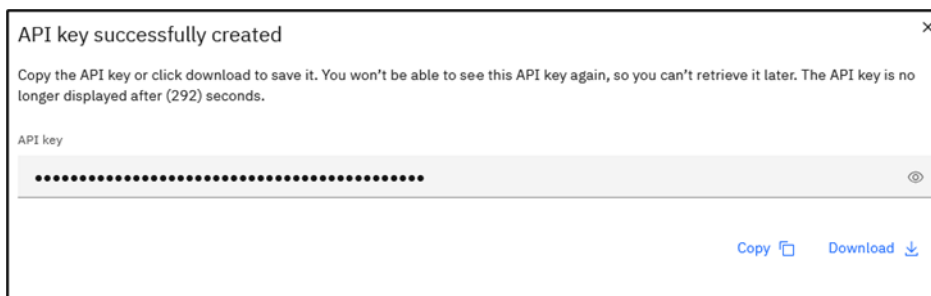
- A default **watsonx.ai endpoint URL** will be displayed for the Dallas region. Ensure the region is always set to **Dallas** at the top right of the watsonx.ai home page.



- Select the **Create API key** button. A **Create API key** widget will be displayed. Enter a name, provide optional description and choose the “Disable the leaked key” option. Click the **Create** button.



- An API key will be created successfully. Copy the API key and save it safely to use for calling the API/SDK. You can also download and save the file in a secure path in your system.



watsonx.ai programmatic options

There are multiple options to help you get started using watsonx.ai APIs/SDKs.

Option 1: Prompt Code on Prompt Lab

Refer to the [access prompt code instructions](#) to learn how to quickly get access to the text generation API within the watsonx.ai Prompt Lab.

Option 2: Different watsonx.ai API capabilities

Explore and leverage different watsonx.ai API capabilities in your solution.

- [Chat](#)
- [Agent-driven chat](#)
- [Tool calling](#)
- [Text generation](#)
- [Time series](#)
- [Text rerank](#)
- [Embeddings](#)
- [Text extraction](#)

Refer to supported API functionality by AI model [here](#).

Access the prompt code (API) from Prompt Lab editor

To prompt an AI model programmatically, you can view and copy the prompt code by selecting the **View code** icon `</>` at the top-right of the prompt lab editor.



The prompt code is available as a Curl, Node.js and Python.

View code ✕

[Create a personal API key](#), and use it to create temporary access tokens. [Learn more](#).

Curl Node.js Python

```
curl "https://us-south.ml.cloud.ibm.com/ml/v1/text/generation?version=2023-05-29" \
  -H 'Content-Type: application/json' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer ${YOUR_ACCESS_TOKEN}' \
  -d '{
    "input": "<start_of_role>system<end_of_role>You are Granite, an AI language model developed by IBM in 2024. You are a cautious assistant. You carefully follow instructions. You are helpful and harmless and you follow ethical guidelines and promote positive behavior.<end_of_text>\n<start_of_role>assistant<end_of_role>",
    "parameters": {
      "decoding_method": "greedy",
      "max_new_tokens": 900,
      "min_new_tokens": 0,
      "stop_sequences": [],
      "repetition_penalty": 1
    }
  }'
  -m "model_id": "ibm/granite-3-8b-instruct",
  "project_id": "30e"
}
```

You will require an IAM access token to authorize the prompt code and need to replace **`${YOUR_ACCESS_TOKEN}`** placeholder in the prompt code. You can create an IAM access token using an API key.

- **API key:**

Refer to [Developer access information](#) to get an API key.

- **Generate IAM Access Token:**

Programmatically generate an IAM access token with the API key using the following cURL command:

```
curl -X POST 'https://iam.cloud.ibm.com/identity/token' -H 'Content-Type: application/x-www-form-urlencoded' -d 'grant_type=urn:ibm:params:oauth:grant-type:apikey&apikey=MY_APIKEY'
```

- **curl -X POST** → Specifies an HTTP **POST** request.
- **URL ("https://iam.cloud.ibm.com/identity/token")** → The endpoint to request an authentication token from IBM Cloud.
- **-H "Content-Type: application/x-www-form-urlencoded"** → Sets the request header to indicate that the data is sent in form-encoded format.
- **-d (Data Payload)** → Sends the required data:
- **grant_type=urn:ibm:params:oauth:grant-type:apikey** → Specifies the OAuth grant type as API Key.
- **apikey=MY_IBM_CLOUD_API_KEY** → Replace MY_IBM_CLOUD_API_KEY with your actual IBM Cloud API key.

Expected Response:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ5IiwiaWF0IjoiMTY3MzE4ODM1MyJ9",
  "refresh_token": "not_supported",
  "token_type": "Bearer",
  "expires_in": 3600,
  "expiration": 1473188353,
  "scope": "ibm openid"
}
```

Note: An IAM token is valid for up to 60 minutes, and it is subject to change. When a token expires, you must generate a new one. Use the property *“expires_in”* for the expiration of the IAM token that you have just created.

watsonx.ai AI agent libraries and tutorials

Explore the watsonx.ai supported AI agent framework libraries and tutorials to help you get started building your AI agent solution.

- [LangChain](#)
- [LangGraph](#)
- [LlamaIndex](#)
- [CrewAI](#)
- [BeeAI](#)
- [AutoGen](#)
- [Python SDK](#)

- [Node.js SDK](#)

Quick start hands-on exercises

Try the quick start exercises and notebooks for sample use cases to get started with using watsonx.ai.

Important notes:

- Refer to [developer access information](#) section to use watsonx.ai credentials as you try the exercises.
- Some of the exercises could include the usage of old model version. You can replace them with newer versions for better performance and output. To check the latest supported AI models on watsonx.ai, either follow [selecting an AI model on Prompt Lab](#) or refer to [supported foundation models on watsonx.ai](#).
- The hackathon-provisioned cloud accounts **do not support solution deployment**. You can run your solution deployment locally on your machine and showcase them in your submissions.
- **Foundation model inferencing** consumes tokens, which are measured as Resource Units (RUs). **1,000 tokens = 1 RU**, and each RU costs **\$0.0001 USD**. [Learn more about tokens and tokenization](#).
- If you are using [Jupyter Notebook editor on watsonx.ai](#), consider selecting a **lower runtime environment** to avoid high resource consumption and quickly depleting your credits. Notebook runtimes are billed based on **Capacity Unit Hours (CUH)** at a rate of **\$1.02 USD per CUH**. [Learn more about capacity unit hours and watsonx.ai Studio pricing plans](#).

watsonx.ai Prompt Lab app templates:

- [LangGraph LLM app template with function calling capabilities](#) (base template)
- [LlamaIndex Workflow LLM app template with function calling capabilities](#) (base template)
- [CrewAI LLM app template with function calling capabilities](#) (base template)
- [arXiv Research agent](#) (community template)
- [Agentic RAG LangGraph template](#) (community template)

BeeAI Agent Framework:

- [BeeAI framework examples](#)

LangChain and LangGraph:

- [Create a LangChain AI Agent in Python using watsonx](#)
- [Build a RAG agent using LangGraph to answer complex questions](#)
- [Build a LangChain agentic RAG system using the Granite model in watsonx.ai](#)
- [Use watsonx, and LangChain Agents to perform sequence of actions](#)
- [Use watsonx, and LangChain to make a series of calls to a language model](#)

- [arXiv Research agent](#)
- [Base LangGraph LLM app template with function calling capabilities](#)

LlamaIndex:

- [Use watsonx and LlamaIndex for Text-to-SQL task](#)
- [Use watsonx, and `llama-3-1-70b-instruct` and LlamaIndex to make simple chat conversation and tool calls](#)
- [LlamaIndex Workflow LLM app template with function calling capabilities](#)

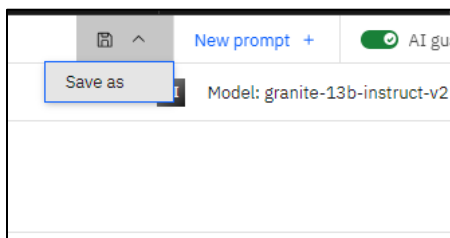
CrewAI:

- [Leveraging CrewAI and IBM watsonx](#)
- [Build an agentic framework with CrewAI memory, i18n, and IBM watsonx.ai](#)
- [Base CrewAI LLM app template with function calling capabilities](#)

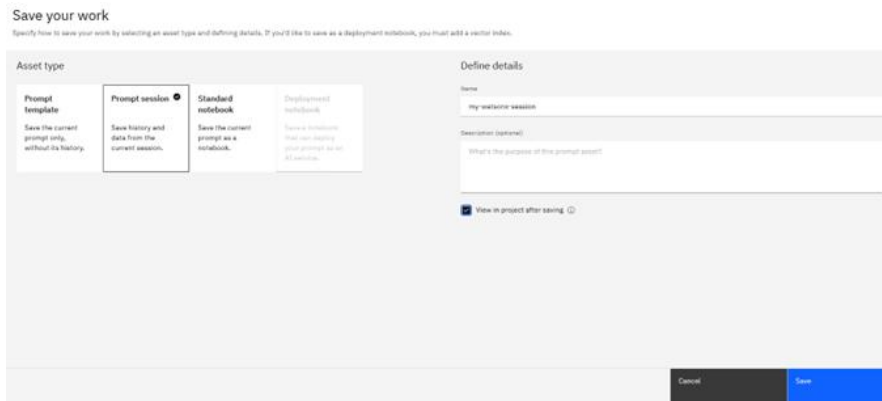
Save your Prompt Lab session

You can save your Prompt Lab editor session for later use.

1. At the top of the Prompt Lab screen, select the **Save work** dropdown button and then select the **Save as** option.



2. A **Save your work** widget will appear. Select **Prompt session** under the **Asset type** option.
3. Enter a **name** and check the **View in project after saving** option under the **Define details** section.
4. Finally, click the **Save** button. Once you save, you will see the saved work under the **Assets** tab



You can also save your work as:

- **Prompt template** to save only the current prompt without its history and selecting a **Task** suitable for your prompting.
- **Notebook** to continue prompting on a Jupyter Notebook environment. Prior knowledge of notebooks and Python programming language would be helpful to work with a Jupyter notebook. [Read more about notebooks.](#)

Save your work on watsonx.ai

Make sure to save any work you want to retain for your records. IBM Cloud accounts will be deactivated at the end of the hackathon. Follow the steps below to save your work:

1. Go to your project's 'Overview' tab.
2. Select the 'Export or import project' drop down below the Bell icon in the top menu bar.
3. Click the 'Export project' option. This will open 'Export project to desktop' screen.
4. Select all the assets shown in your project (Work saved as Project session cannot be exported) and click 'Export' on the bottom-right of the screen.
5. The next screen will ask for confirmation that all sensitive information has been removed.
6. Click on 'Continue export'.
7. The download (zip) will be initiated, and the file will be saved on your computer.

Appendix: Example use cases

You are not limited to these ideas, but here are several examples for how you could apply IBM Bob, with optional use of watsonx Orchestrate and watsonx.ai, to solve real challenges in this hackathon:

- **Code understanding and onboarding accelerator:** Create a solution where IBM Bob analyzes existing code repositories and explains what the code does, how components interact, and where critical logic lives. Optionally, watsonx.ai can be used to summarize insights in plain language for different audiences, while watsonx Orchestrate could coordinate follow up tasks such as generating onboarding checklists or learning guides for new team members.
- **Automated documentation and test companion:** Build a solution where IBM Bob reviews application logic and automatically generates documentation and relevant tests to improve reliability and clarity. Optionally, watsonx.ai can help tailor documentation tone or structure for different stakeholders, and watsonx Orchestrate could manage workflows that keep documentation and tests updated as code evolves.
- **Repetitive task and boilerplate reduction assistant:** Develop a solution that uses IBM Bob to reduce time spent on repetitive development tasks such as generating boilerplate code, updating configurations, or refactoring common patterns. Optionally, watsonx Orchestrate could coordinate related activities such as opening tasks, tracking progress, or applying changes consistently across multiple projects.
- **Intelligent debugging and issue insight solution:** Create a solution where IBM Bob assists with debugging by explaining execution flow, highlighting potential root causes, and summarizing changes that may have introduced issues. Optionally, watsonx.ai can enhance insight generation by identifying patterns across incidents, while watsonx Orchestrate could help route findings into issue tracking or reporting workflows.
- **Guided modernization and improvement planner:** Build a solution that uses IBM Bob to assess existing code quality and suggest improvements such as refactoring complex logic or updating outdated patterns. Optionally, watsonx.ai can help score or categorize modernization opportunities, while watsonx Orchestrate can coordinate the sequence of improvement tasks to ensure changes are applied efficiently and consistently.