# Deploying Loan Risk AI Agent on IBM Cloud Code Engine

GitHub repository: <https://github.com/IBM/ai-agent-for-loan-risk>

Table of Contents

[Prerequisites 1](#_Toc195104032)

[Deployment Steps 3](#_Toc195104033)

[1. Confirm prerequisites and gather information 3](#_Toc195104034)

[2. Create a Project in watsonx.ai 3](#_Toc195104035)

[3. Create a namespace in Container Registry 6](#_Toc195104036)

[4. Deploy application to Code Engine 6](#_Toc195104037)

[5. Validate the application. 11](#_Toc195104038)

[6. Enhance application with additional optional features. 11](#_Toc195104039)

[Optional Features 12](#_Toc195104040)

# Prerequisites

Requires IBM Cloud account with:

* watsonx.ai services ([refer](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/signup-wx.html?context=wx&audience=wdp)). This includes:
  + watsonx.ai Runtime
  + watsonx.ai Studio
  + Cloud Object Storage
* Account user with administrative privileges for:
  + watsonx.ai – to access services, create Projects for storing assets, get Project ID etc. ([refer](https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/projects.html?context=wx&audience=wdp))
  + Code Engine – to create Project and deploy the application. ([refer1](https://cloud.ibm.com/docs/codeengine?topic=codeengine-getting-started) [refer2](https://cloud.ibm.com/docs/codeengine?topic=codeengine-app-source-code))
  + Container Registry – to create namespace, store images for deploying the application (
* Account user’s API key ([refer](https://cloud.ibm.com/docs/account?topic=account-userapikey&interface=ui#create_user_key))

# Deployment Steps

Note on costs: The deployment will incur costs for Code Engine runtime, watsonx.ai runtime and watsonx.ai LLM inferencing.

## Confirm prerequisites and gather information

* 1. Note the user’s API key value. It will be needed for creating container registry secret. It will also be the environment variable WATSONX\_AI\_APIKEY.
  2. Note the region where watsonx.ai is deployed. Get the service URL API endpoint for your region from [this list](https://cloud.ibm.com/apidocs/watsonx-ai#endpoint-url)  - e.g., for us-south/Dallas region it is <https://us-south.ml.cloud.ibm.com>. This will be the environment variable WATSONX\_SERVICE\_URL.
  3. Note/decide the region to deploy the application resources, e.g. us-south, use-east, eu-de etc.
  4. Note/decide/create the resource group to use in the account, e.g. agentic-ai-rg

## Create a Project in watsonx.ai

(Or use an existing one and capture its Project Id)

To create a new Project, you will need to configure the Object Storage. Associate the Project with watsonx.ai Runtime service.

Detailed steps for reference:

2.1. Create a Project. Also define the Cloud Object Storage for storage.

<https://dataplatform.cloud.ibm.com/docs/content/wsj/getting-started/projects.html?context=wx&audience=wdp>  
<https://dataplatform.cloud.ibm.com/projects/?context=wx>

A screenshot of a black screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

2.2. From the Manage tab > Services and integrations >Associate service, - Associate it with the watsonx.ai Runtime service.

A screenshot of a computer

AI-generated content may be incorrect.

2.3. From the Manage > General tab, capture the Project ID

A screenshot of a computer

AI-generated content may be incorrect.

2.4. Note the Project Id. This will be the environment variable WATSONX\_PROJECT\_ID.

## Create a namespace in Container Registry

Detailed steps for reference:

<https://cloud.ibm.com/containers/registry/namespaces>

Name: agentic-ai-app-namespace

A screenshot of a computer

AI-generated content may be incorrect.

## Deploy application to Code Engine

Detailed steps for reference:

Refer section [Deploying your app from repository source code from the console](https://cloud.ibm.com/docs/codeengine?topic=codeengine-app-source-code&interface=ui)

Below is some information you will need to complete the steps.

* 1. Click Create Project
  + Project name: agentic-ai-app-project, Select region and resource group.
  + Component type. Select Application. Application name: agentic-ai-app
  + Code. Select Build Container image from source.
  + Code repo URL: <https://github.com/IBM/ai-agent-for-loan-risk.git>

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Specify build details.

Mostly use defaults. But create new registry secret.

A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

* 1. Create registry secret.

Name agentic-ai-app-secret.   
You will also need to select region, enter the user’s API key, and select the namespace that was created earlier.

A screenshot of a login page

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Once done with Specify build details, this is what the Code section will look.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Scroll down to optional settings and add environment variables.
     + WATSONX\_AI\_APIKEY
     + WATSONX\_PROJECT\_ID
     + WATSONX\_SERVICE\_URL

A screenshot of a computer

AI-generated content may be incorrect.

* 1. Create/deploy the application. Check status



A screenshot of a computer

AI-generated content may be incorrect.

* 1. Click on Test application (right top) > Application URL to launch application.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Validate the application.

Ask “What is the interest rate for matt? Explain how it was determined?”. You should get a response explain high risk and 8% interest rate.

Refer to the application [usage section](https://github.com/IBM/ai-agent-for-loan-risk/blob/main/artifacts/usage-examples/usage-examples-README.md).

## Enhance application with additional optional features.

Once initial application is deployed and running successfully, you can add enhancements.

* Using RAG LLM (Agentic RAG feature)
* Using watsonx Assistant/Orchestrate (Chat widget)

Refer to Optional Features section.

# Optional Features

Once initial application is deployed and running successfully, you can enhance with some additional optional features.

* Using RAG LLM (Agentic RAG feature)
* Using watsonx Assistant/Orchestrate (Chat widget)

#### **Using RAG LLM (Agentic RAG feature):**

By default, the risk and interest rate tools of the AI agent simulate risk and interest rate determination. By adding this feature, the AI agent tool will make RAG query to retrieve relevant content from the bank documents and use that to determine the risk and interest rate. In the AI agent response you will be able to see the content from the document table and its interpretation by the AI agent.

* Create a new watsonx.ai Deployment Space with Deployment stage set as Production. [(refer)](https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ml-spaces_local.html?context=wx&locale=en&audience=wdp)
* Create a vector index asset in the watsonx.ai Project using these [content PDF documents](https://github.com/IBM/ai-agent-for-loan-risk/blob/main/artifacts/deployment/artifacts/data). To create the index use vector store - "In memory", embedding model - "allminilm-l6-v2", chunksize - "2000", chunk overlap - "200".
* Open the vector index you just created in Prompt Lab and set the generative AI model to "mistral-large". Test the index by asking some questions e.g., what is the risk for credit score 655 and account status closed?, what is the interest rate for medium risk?, and confirm answers are using RAG from the content PDF documents.
* Deploy the vector index as watsonx.ai Deployment on AI service for inferencing using the "fast path" option (use the "Deploy" button). [(refer-1)](https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ai-services-overview.html?context=wx&locale=en) [(refer-2)](https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ai-services-prompt-lab.html?context=wx) and [(refer-3)](https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/ai-services-deploy-fast-path.html?context=wx).
* Capture the watsonx.ai Deployment private endpoint for the vector index for RAG inferencing (use non-stream; ends with ai\_service?version=...)
* On Code Engine add the following environment variables with the values captured above and redeploy the application (ENABLE\_RAG\_LLM=true and WATSONX\_RISK\_RAG\_LLM\_ENDPOINT=endpoint captured above)
* The application will now use the RAG content for risk and interest tools.

#### **Using watsonx Assistant/Orchestrate (Chat widget):**

By adding this feature, you can get a more conversational/chat experience when asking questions in the watsonx Assistant chat widget. The conversation is single turn and the watsonx Assistant skills can be enhanced further if needed.

* Note the Code Engine URL for the deployed application.
* Open API file (agentic-ai-app-custom-ext-openapi.json) and update the URL with the deployed applicaiton URL.
* Create an action skill in watsonx Assistant instance
* Create and add a custom extension by importing the updated Open API file (agentic-ai-app-custom-ext-openapi.json).
* Import the zip file to set up the actions that use the custom extension (wx-asst-agentic-ai-app.zip)
* Open the watsonx Assistant Web chat configuration and note the integrationID, region and serviceInstanceID from the Embed script tab.
* On Code Engine open the deployed application configuration, add the following environment variables with the values captured above and redeploy the application (ENABLE\_WXASST=true, WXASST\_INTEGRATION\_ID, WXASST\_REGION, WXASST\_SERVICE\_INSTANCE\_ID captured above)
* The watsonx Assistant will become available on the page <application-url>/wx.html