

Setup for the Retail use case lab

- [Setup for the Retail use case lab](#)
 - [Introduction](#)
 - [Virtual machine option](#)
 - [Lab materials](#)
 - [Local machine](#)
 - [VS Code](#)
 - [Virtual machine](#)
 - [VS Code](#)
 - [Environments](#)
 - [watsonx.ai](#)
 - [The .env file](#)
 - [API key](#)
 - [Entitlement key](#)
 - [watsonx Orchestrate ADK](#)
 - [Local machine](#)
 - [Virtual machine](#)
 - [watsonx Orchestrate](#)
 - [Tavily](#)

Introduction

This document contains the documentation for the setup of the environment to prepare for the step-by-step walkthrough of the [Retail use case](#).

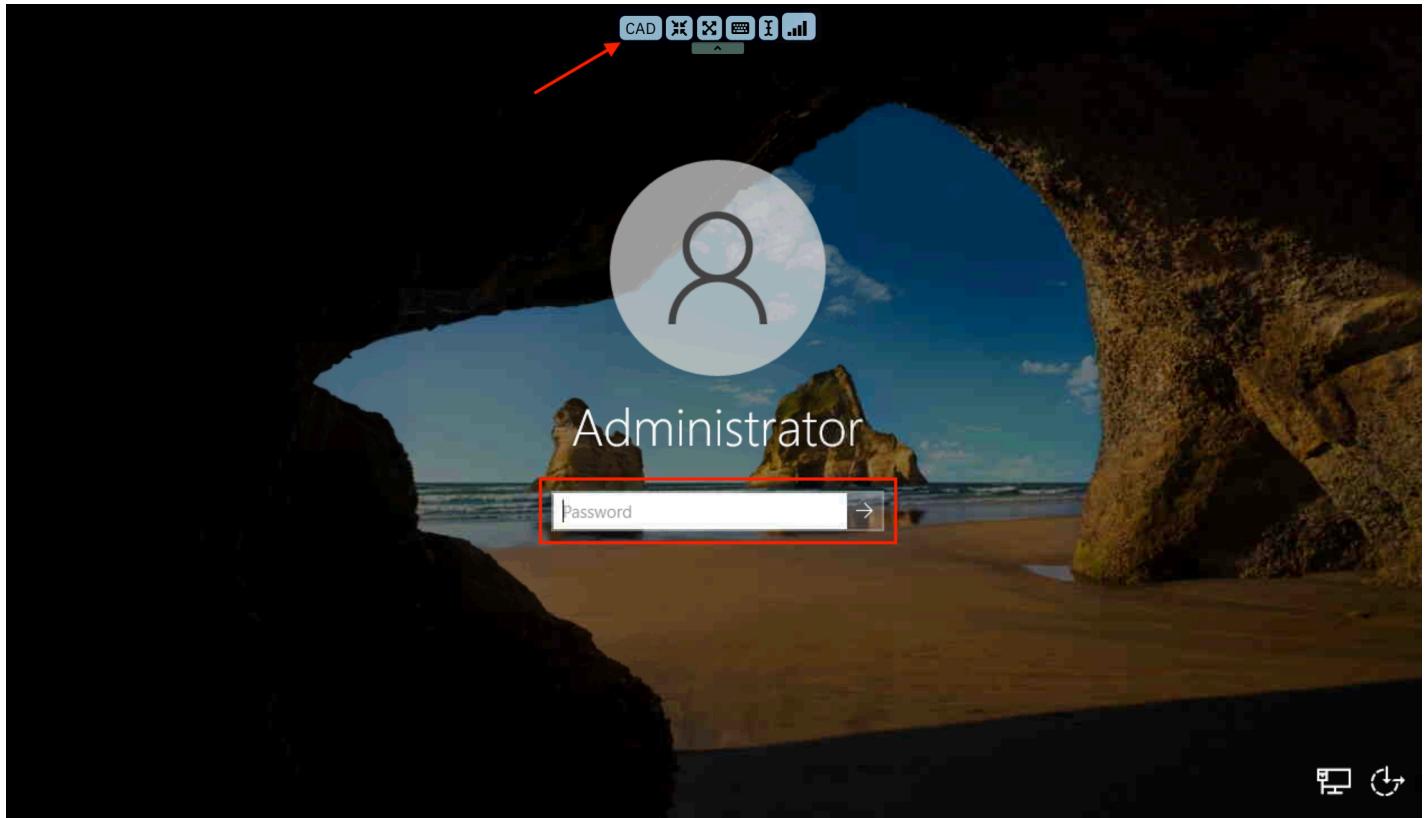
The use case takes you through the creation of tools and agents using the [IBM watsonx Orchestrate Agent Development Kit \(ADK\)](#). This toolkit can be installed on a local machine and brings with it the core components of watsonx Orchestrate, as container images that are running in a container runtime like Docker or Rancher. It also installs a CLI that can be used to manage a locally running instance as well as remote instances running in the cloud.

Virtual machine option

For this lab, you have the option to define and run the agents locally on your own laptop, using the ADK. The system requirements are mentioned/referenced in the [ADK install section](#) below.

If you are unable to run the ADK locally, your instructor will provide you with access to a virtual machine that has the ADK already installed. You can access the VM by using a console link that opens a view of the VM's desktop UI in a browser tab.

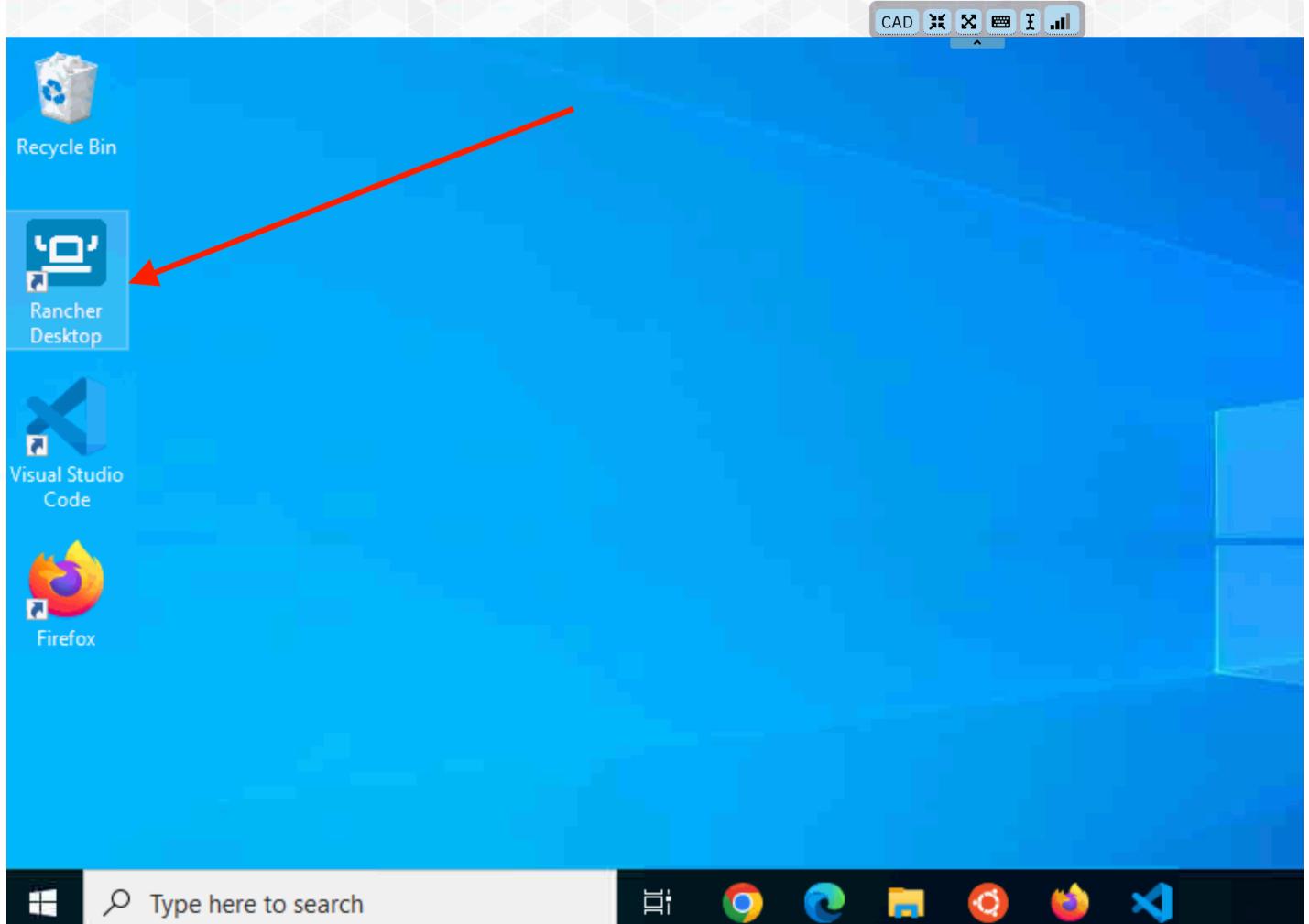
Click the **Ctrl-Alt-Delete CAD** button (annotated with red arrow) to be prompted to enter username and password. Keep the default username as **Administrator** and enter the password **IBMDem0s** in the password field (annotated with red rectangle) and hit **ENTER**.



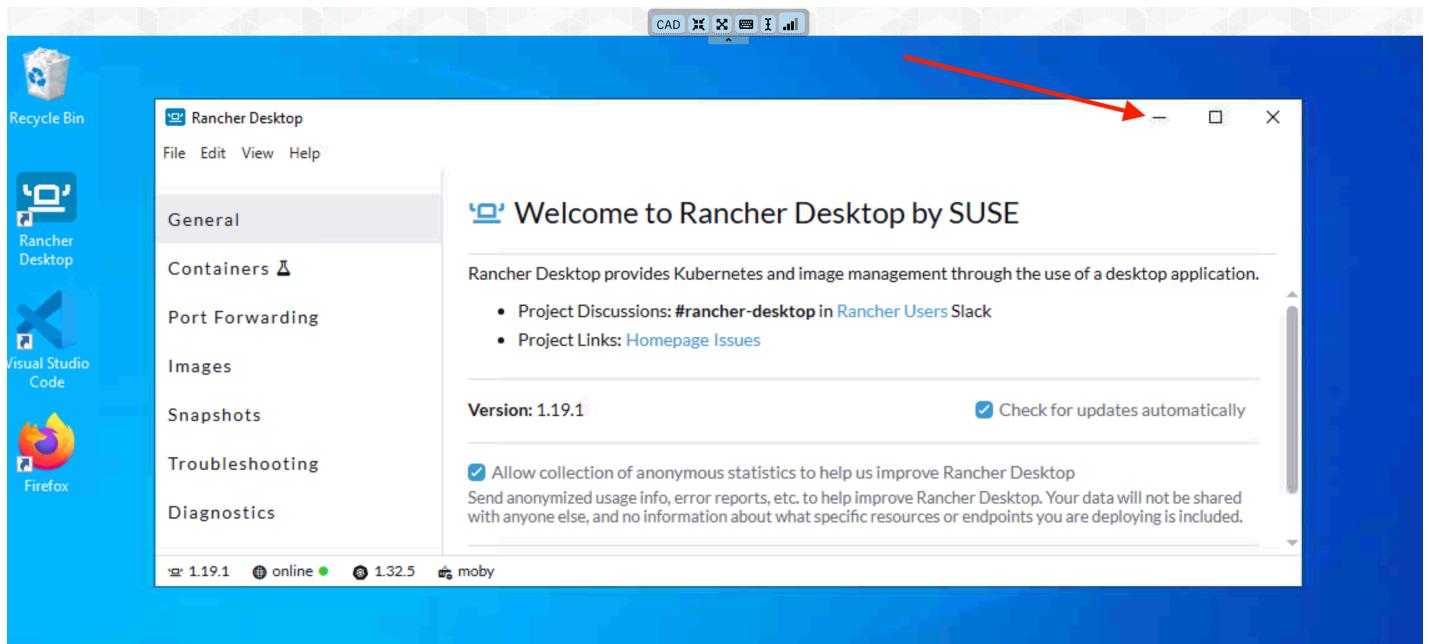
Optional It is also recommended to resize the virtual screen to Full Screen. To do so, click the **Resize** icon (annotated with red arrow) and select **Fullscreen** (annotated with red rectangle).

If you get a pop-up about an Unplanned shutdown, cancel that pop-up.

Once logged into the Windows VM, let's start **Rancher Desktop**. This is the container runtime we will use later for the ADK. On the home screen, double-click the Rancher Desktop icon.

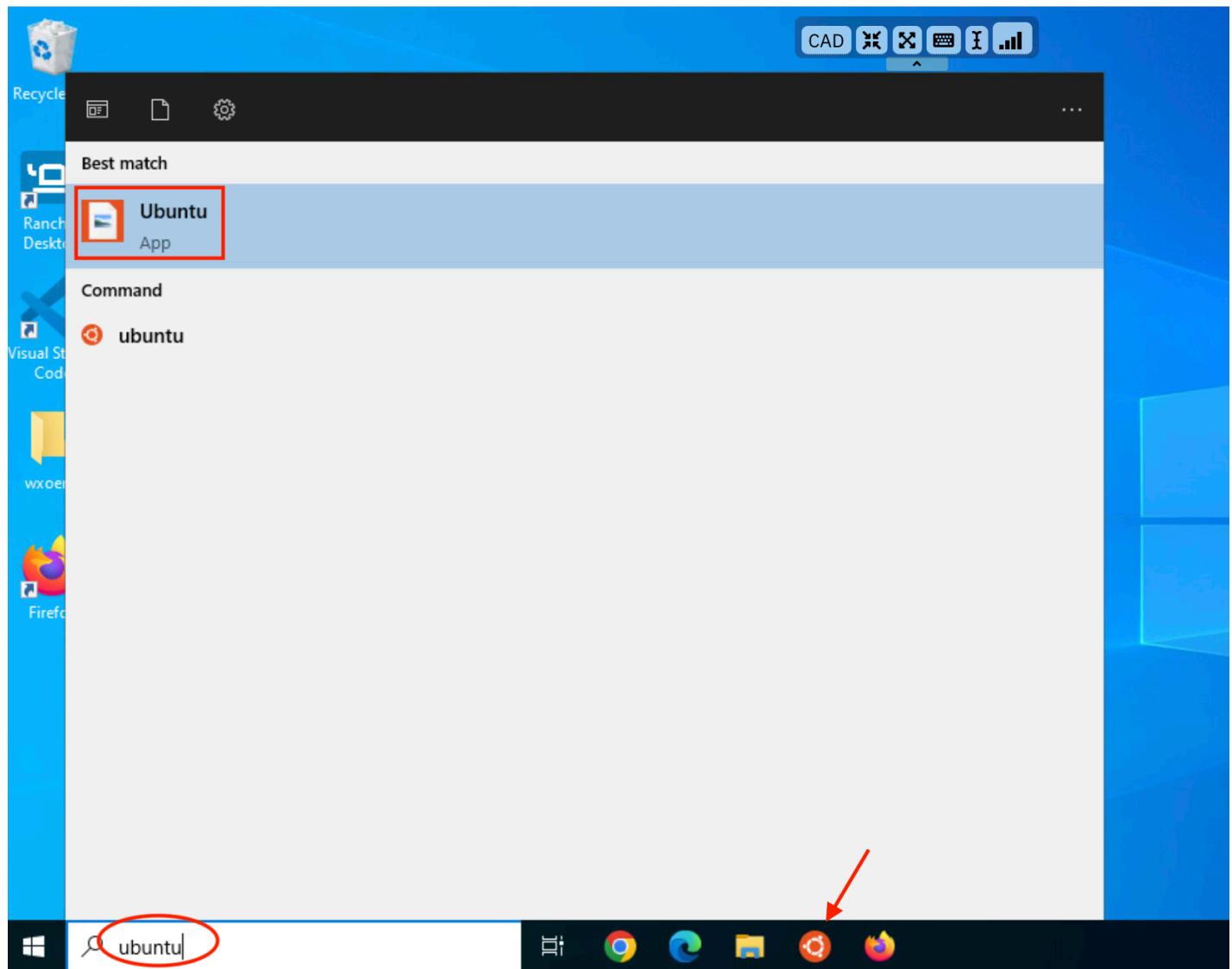


You will see in the Rancher console that it is starting up the service. Once that has completed, you can minimize the window, since we won't need it anymore.



Next, you need to open a command line terminal with access to the WSL instance. To start, click the **Ubuntu** icon (annotated with red arrow) pinned to the taskbar to start the Ubuntu terminal. Alternatively, you can type **ubuntu**

into the search field (annotated with a red oval) and click the **Ubuntu** app (annotated with a red rectangle).



In the Ubuntu terminal, you need to activate the Python environment, which is already setup and has the watsonx Orchestrate ADK pre-installed.

```
source /home/techzone/wsl_wxoenv/bin/activate
```

```
techzone@WIN-36L4N1I4M60: ~
techzone@WIN-36L4N1I4M60:~$ source wsl_wxoenv/bin/activate
(wsl_wxoenv) techzone@WIN-36L4N1I4M60:~$ which python
/home/techzone/wsl_wxoenv/bin/python
(wsl_wxoenv) techzone@WIN-36L4N1I4M60:~$ python -V
Python 3.13.3
(wsl_wxoenv) techzone@WIN-36L4N1I4M60:~$
```

Lab materials

The materials for this lab will be given to you by your instructor in the form of a zip file. You need to unzip this file into a folder on your machine. The file contains a set of markdown files that represent the instructions for various parts of the bootcamp (including this very file), as well as code samples that you are going to use. Where to unzip the file differs based on whether you are running this lab on your local machine or on a virtual machine provided to you.

Local machine

In this case, you can choose any folder as the base for the material. You should use that same folder as the current directory when installing the ADK (which is described in more detail [below](#)). After unzipping, listing the files in a command terminal should look like shown below (this screenshot is taken on MacOS):

```
(wxo_env) Andres-MacBook-Pro-2:agentic-ai-client-bootcamp andretost$ ls -l
total 11200
-rw-r--r--  1 andretost  staff  5729787 May  8 17:58 agentic-bootcamp.png
drwxr-xr-x  9 andretost  staff    288 Jun 11 12:27 environment-setup
-rw-r--r--  1 andretost  staff    973 Jun  9 08:45 README.md
drwxr-xr-x 10 andretost  staff   320 Jun  6 12:59 usecases
(wxo_env) Andres-MacBook-Pro-2:agentic-ai-client-bootcamp andretost$ █
```

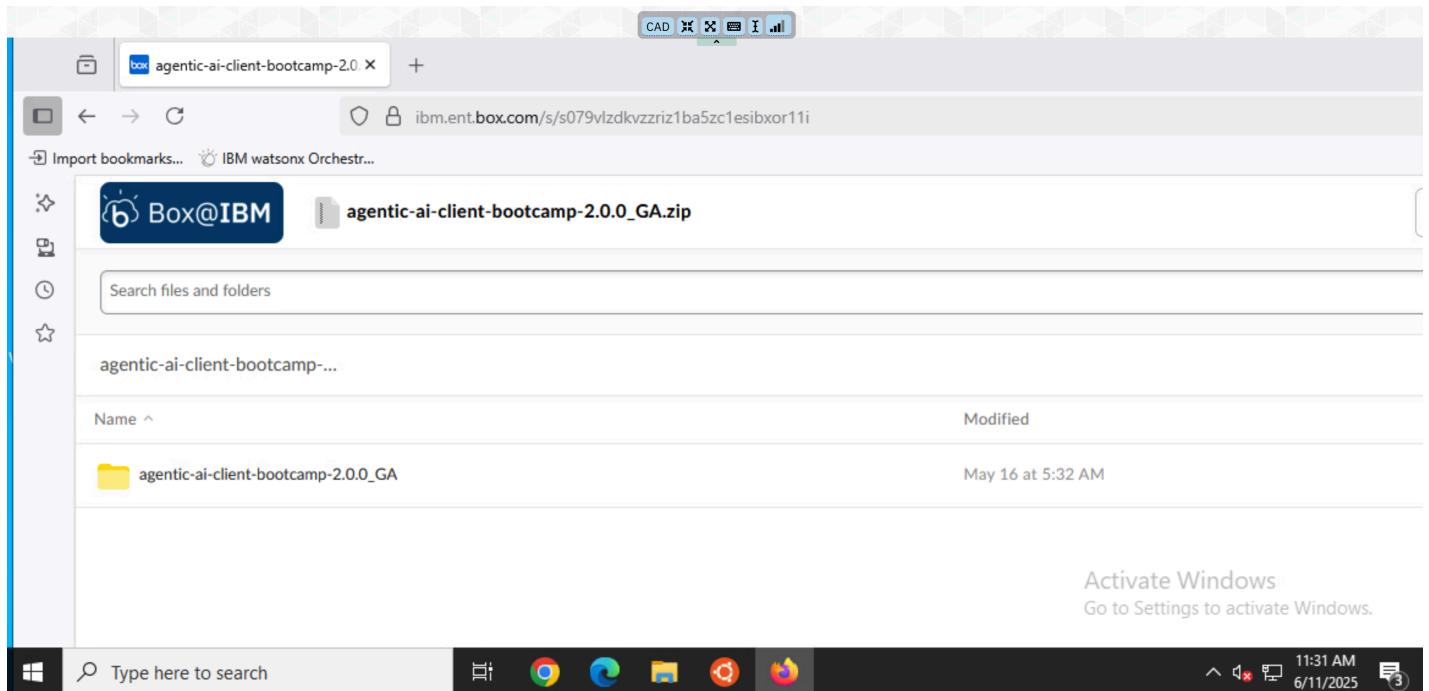
VS Code

We recommend you use VS Code to view the materials and edit files as needed. Assuming you have the command line starter installed, you should be able to start VS Code right from the command line by entering `code .`

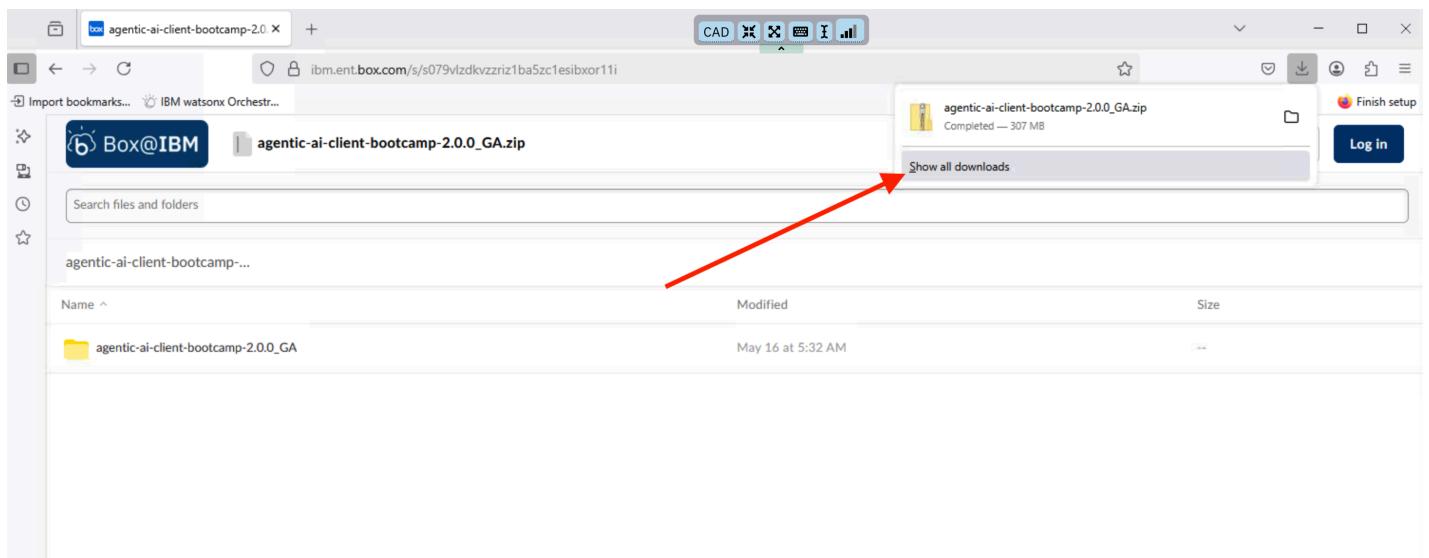
Virtual machine

The virtual machine is using Windows as its operating system; however, we will be using the "Windows Subsystem for Linux (WSL)" to run the ADK. When downloading and unzipping the file with materials, you should put it into the folder that has been precreated.

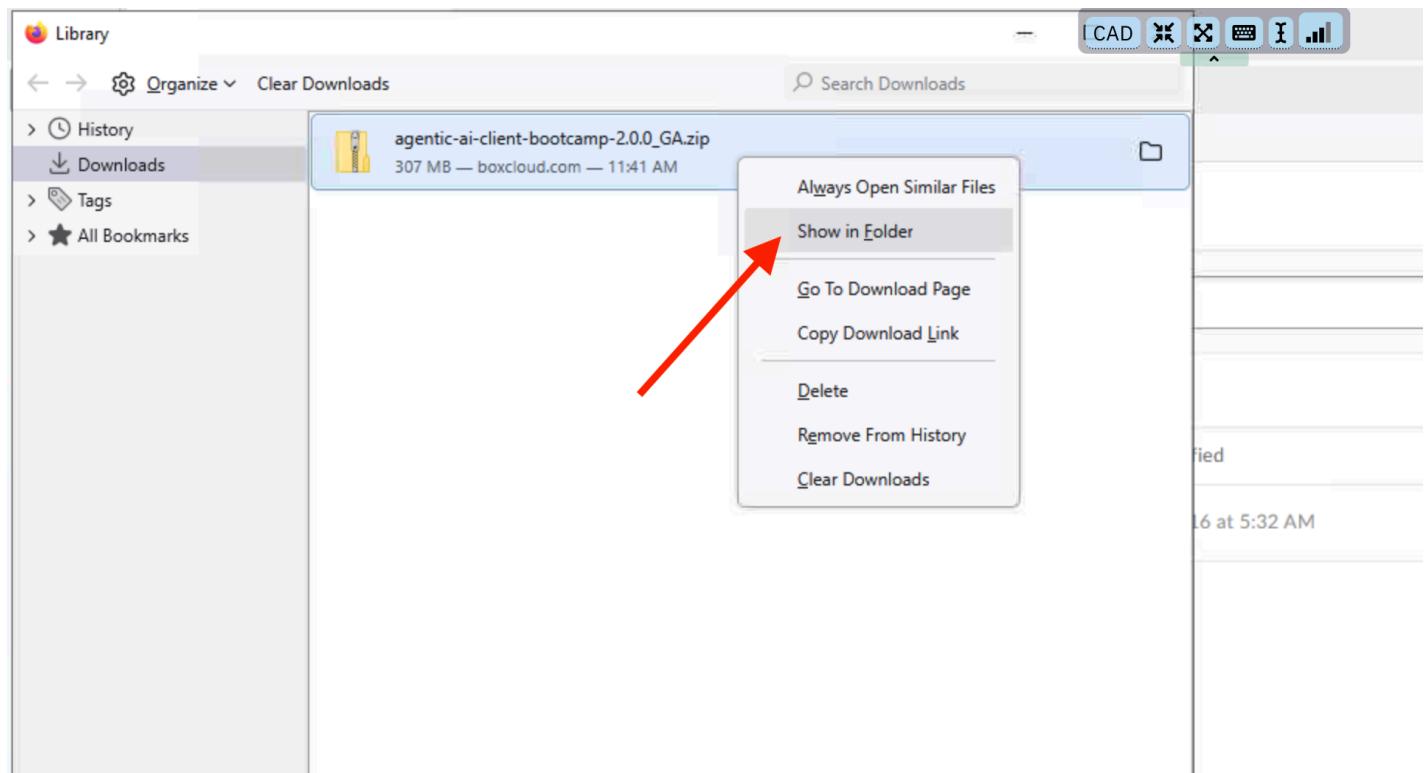
After you open the VM console in your browser, you will see the Windows user interface. There, you can open a Firefox browser window and enter the address your instructor gave you. In the example below, the zip file exists as a downloadable file in Box:



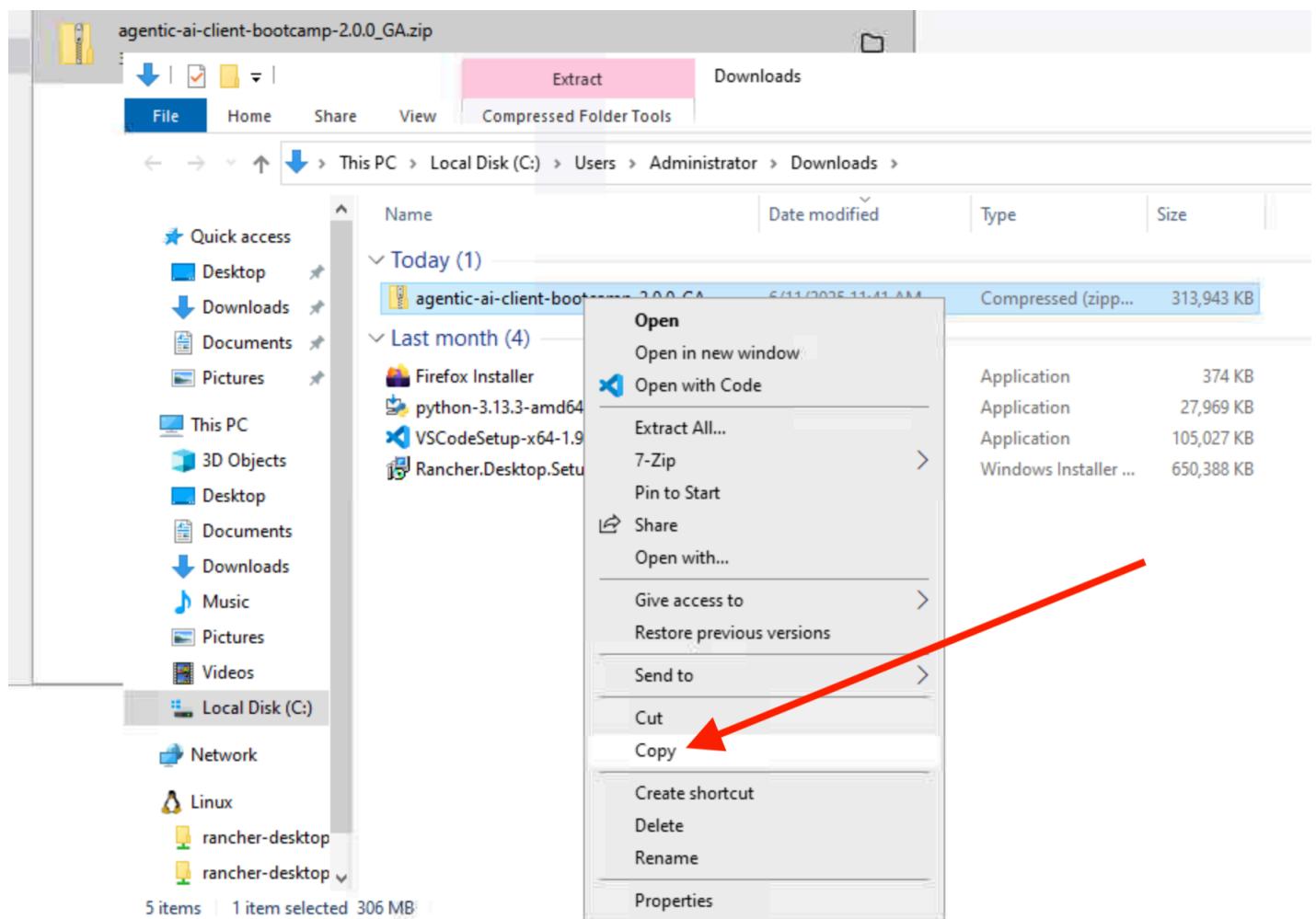
When clicking on the Download button, the file will be downloaded into the Downloads folder on the Windows machine. Open that folder by simply clicking on the Show all downloads button.



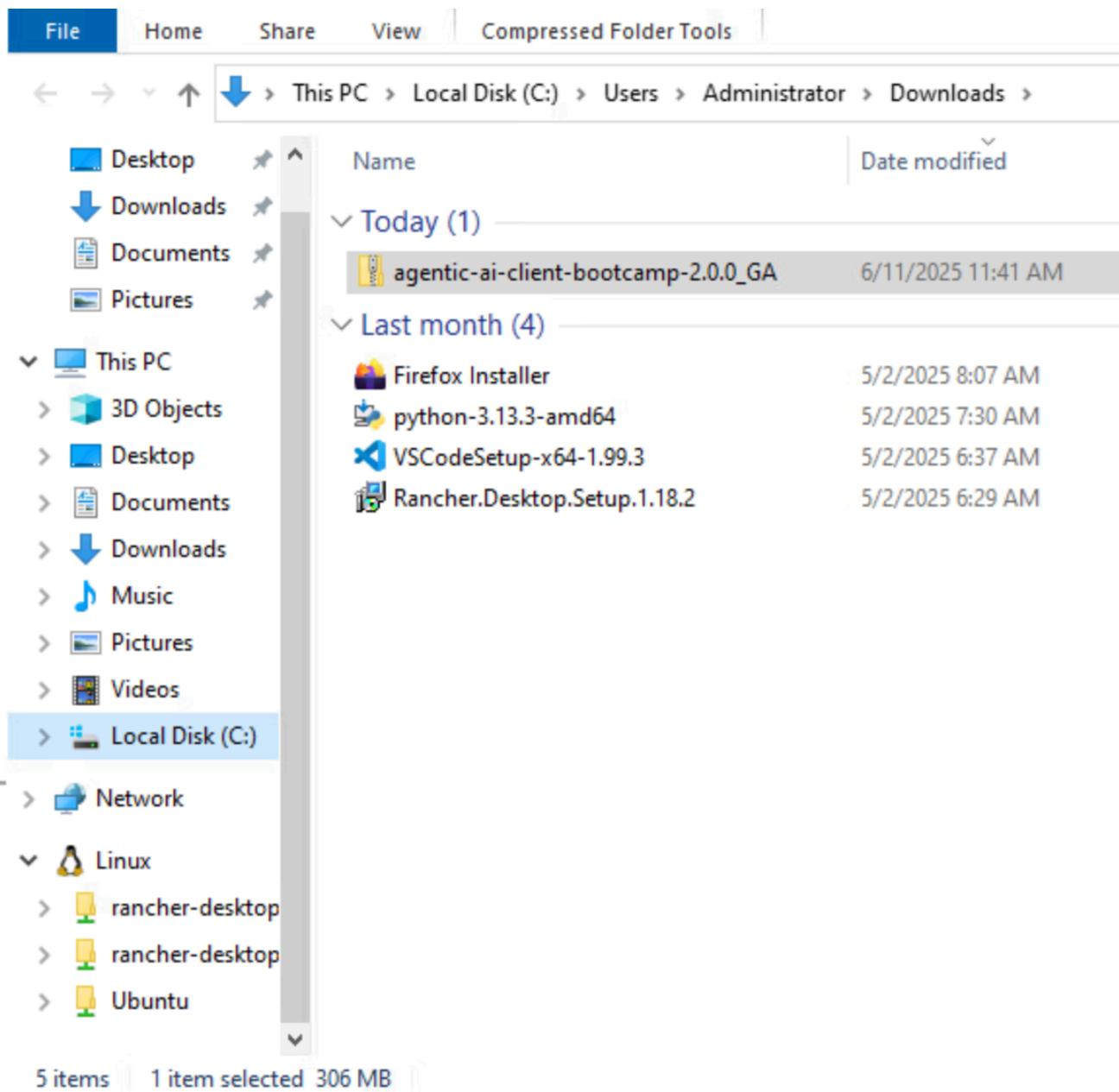
Then click on Show in Folder to open the file explorer window.



Right-click on the zip file and select Copy .



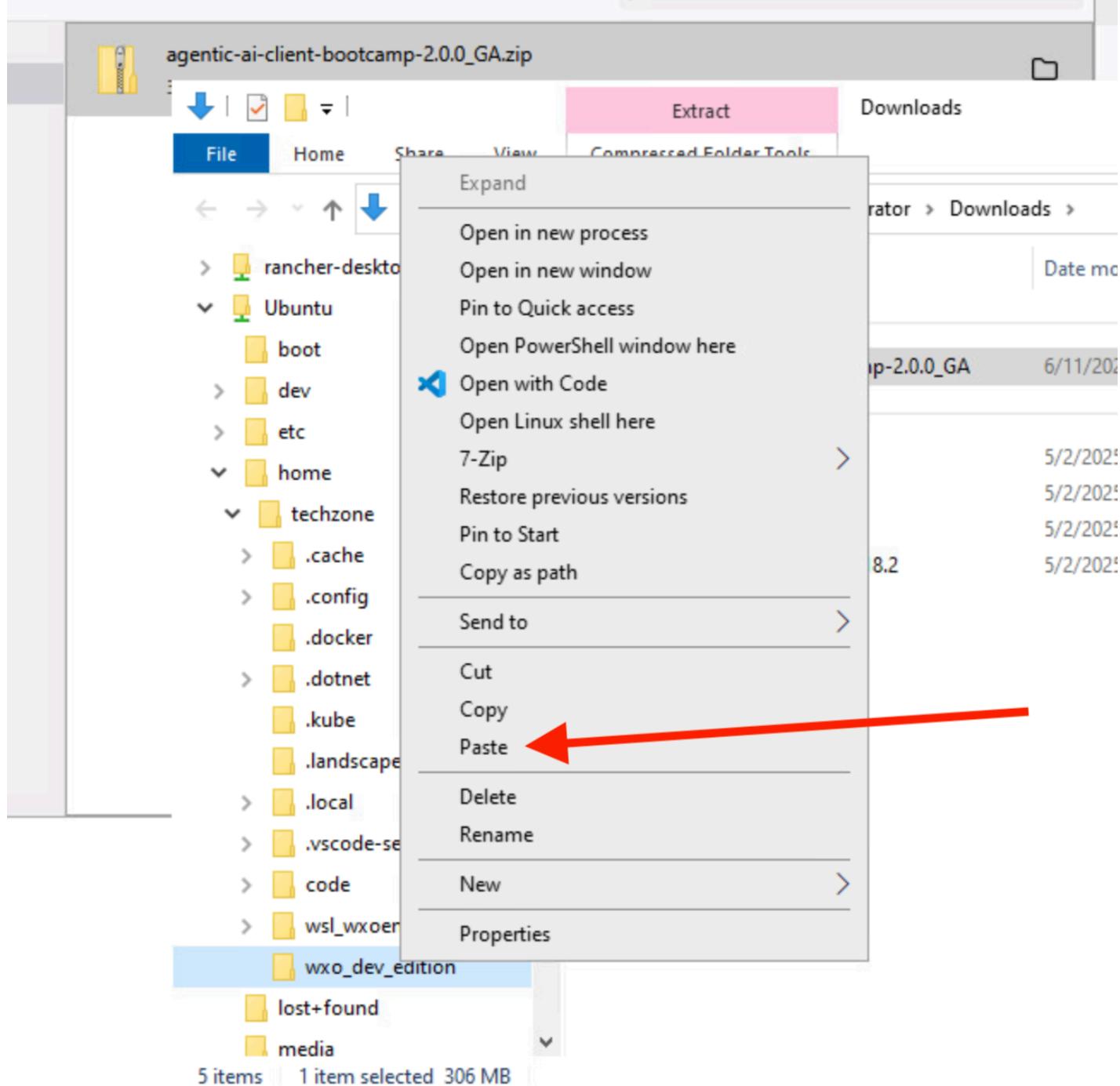
Now navigate to the Ubuntu folder under Linux on the left side of the file explorer menu.



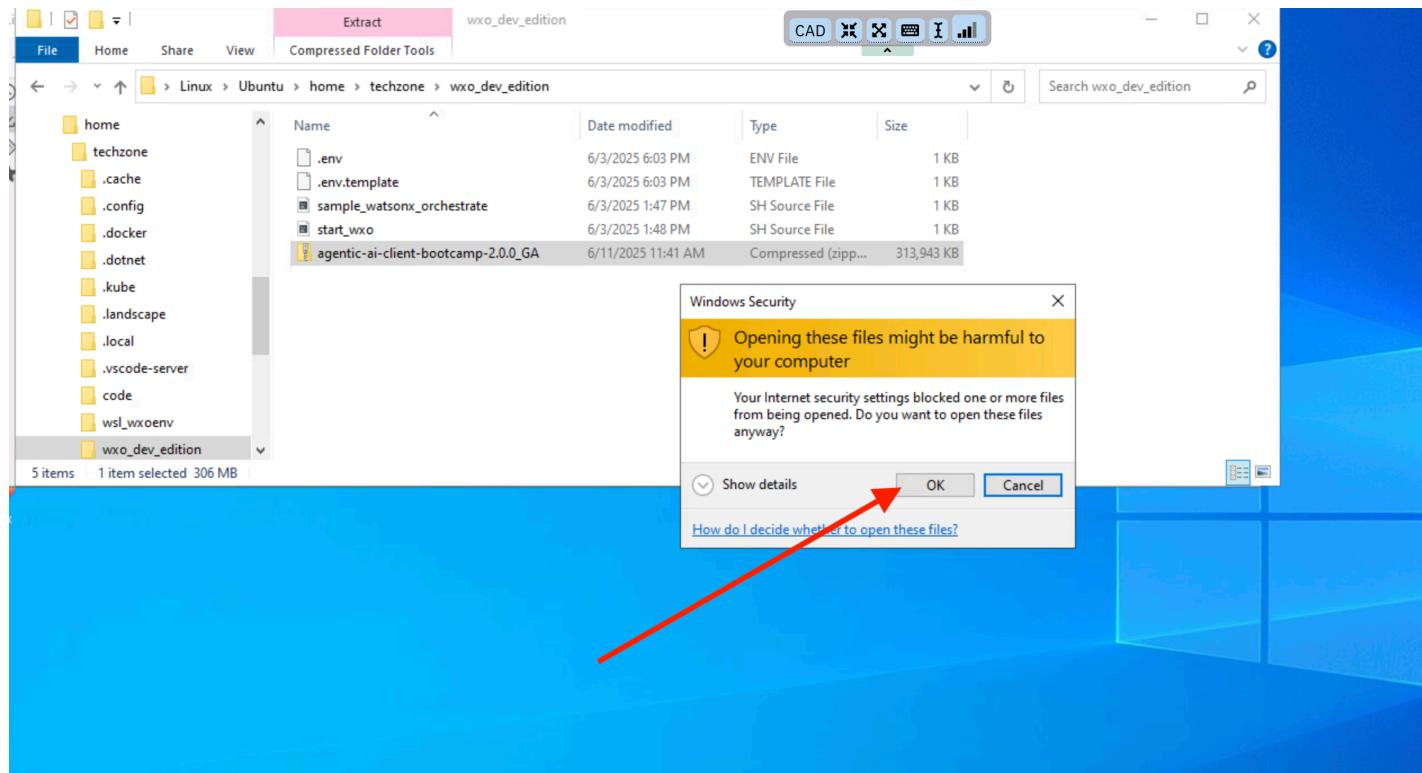
Under the folder, right click on home -> techzone -> wxo_dev_edition and click on Paste.

Clear Downloads

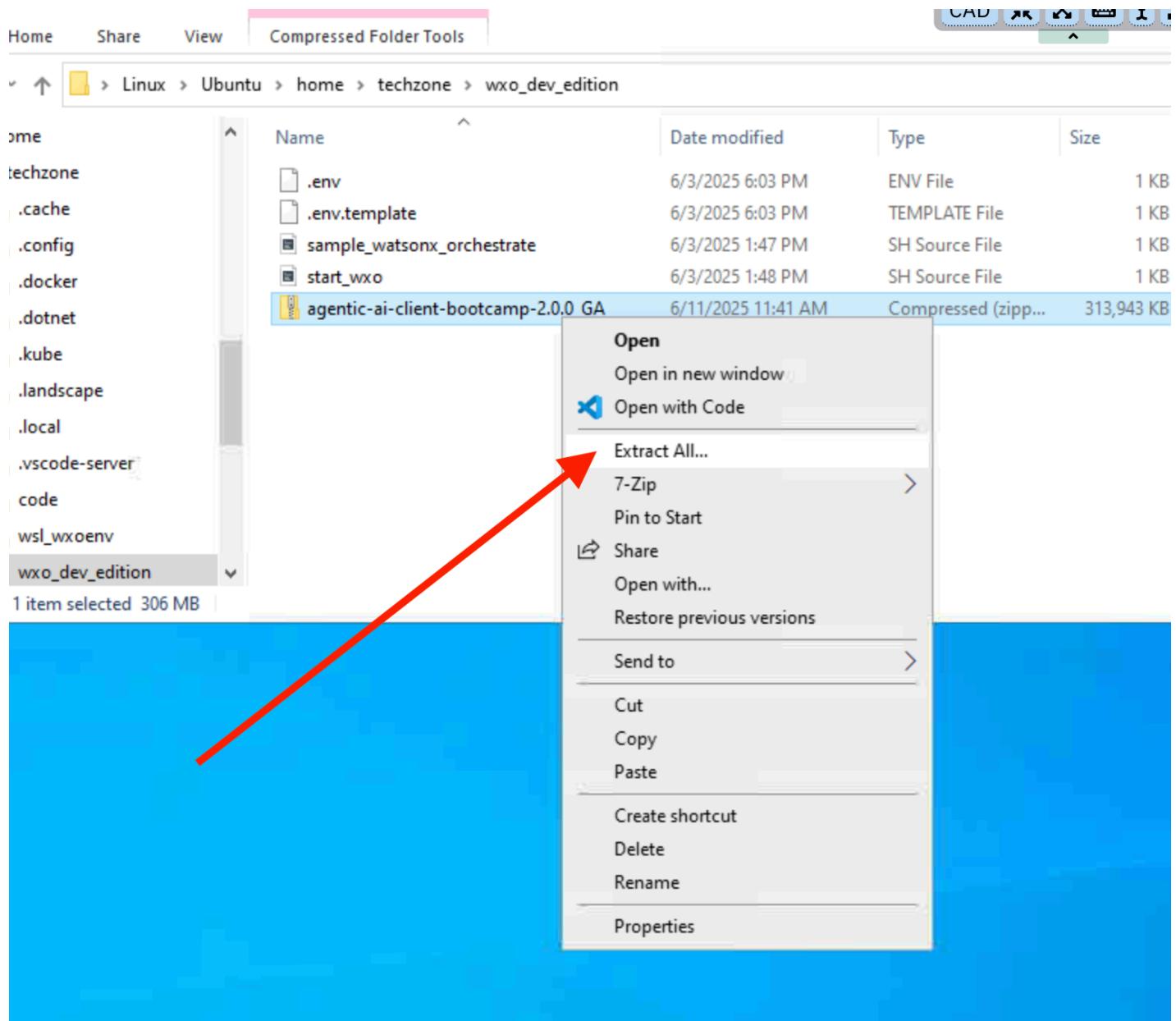
Search Downloads



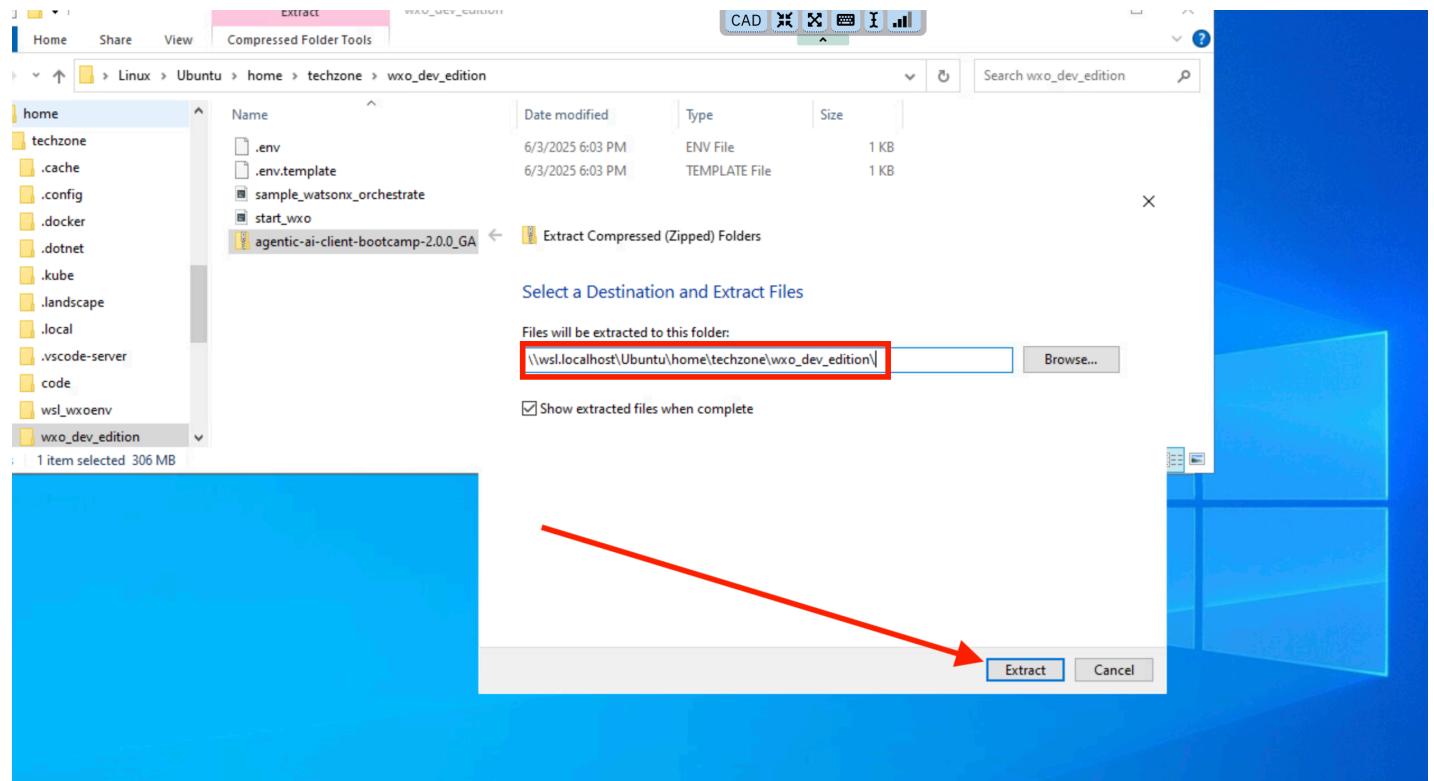
Open the folder and right-click on the zip file. You may receive a warning that this file is from the Internet. Click OK.



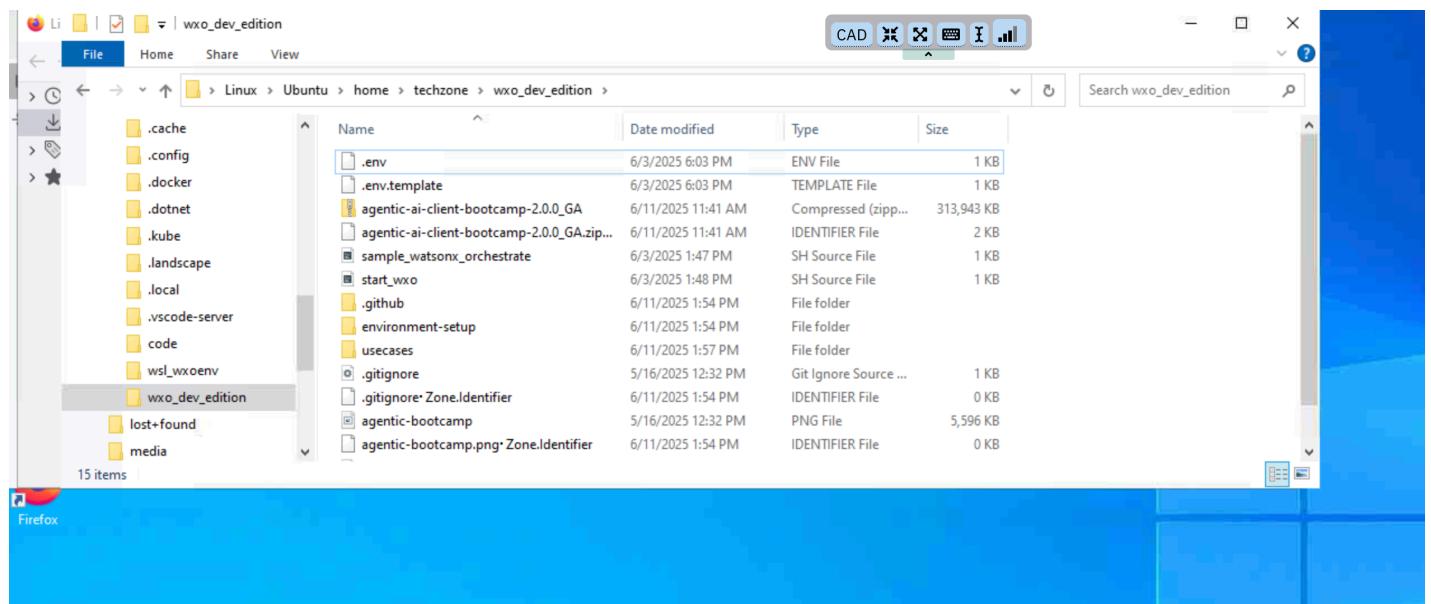
In the context window for the zip file, select Extract all .



As the destination, make sure you enter the `wxo_dev_edition` (which is not the default), as shown below. Then click on `Extract`.

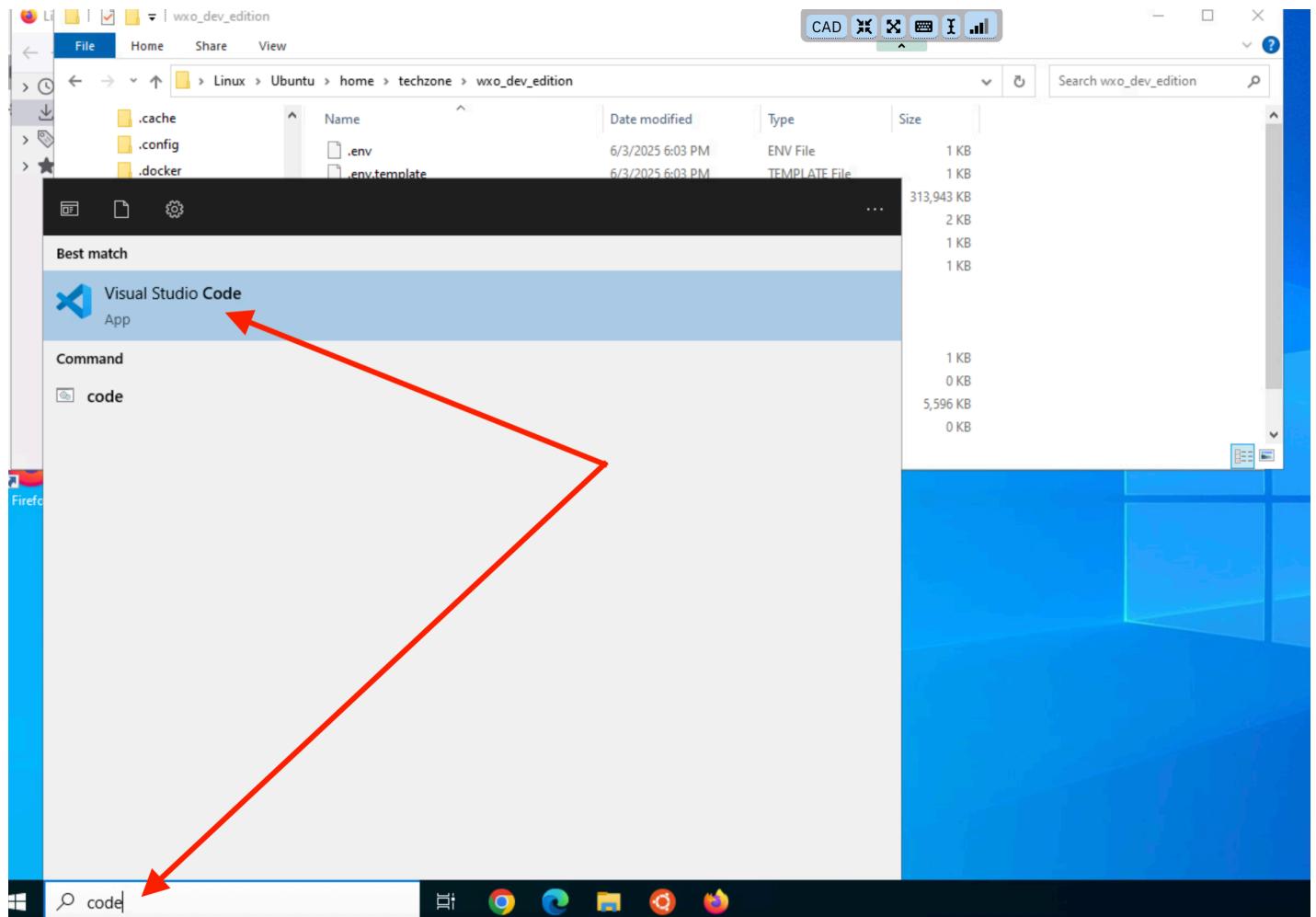


Note that the extraction process can take a couple of minutes. After it completes, your file explorer window should show the extracted files in the `wrxo_dev_edition` folder. If they are placed into a subfolder, you can cut and paste them.

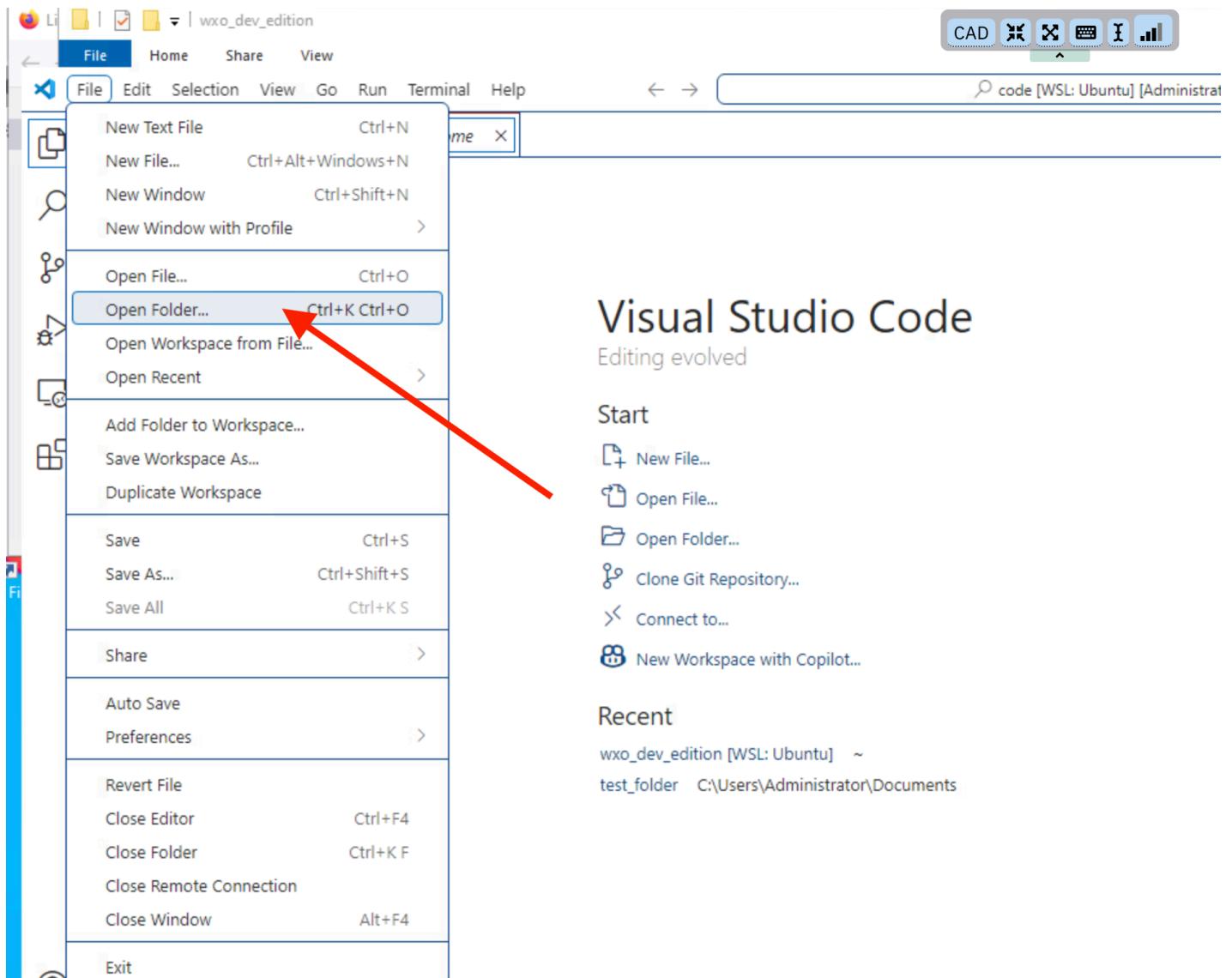


VS Code

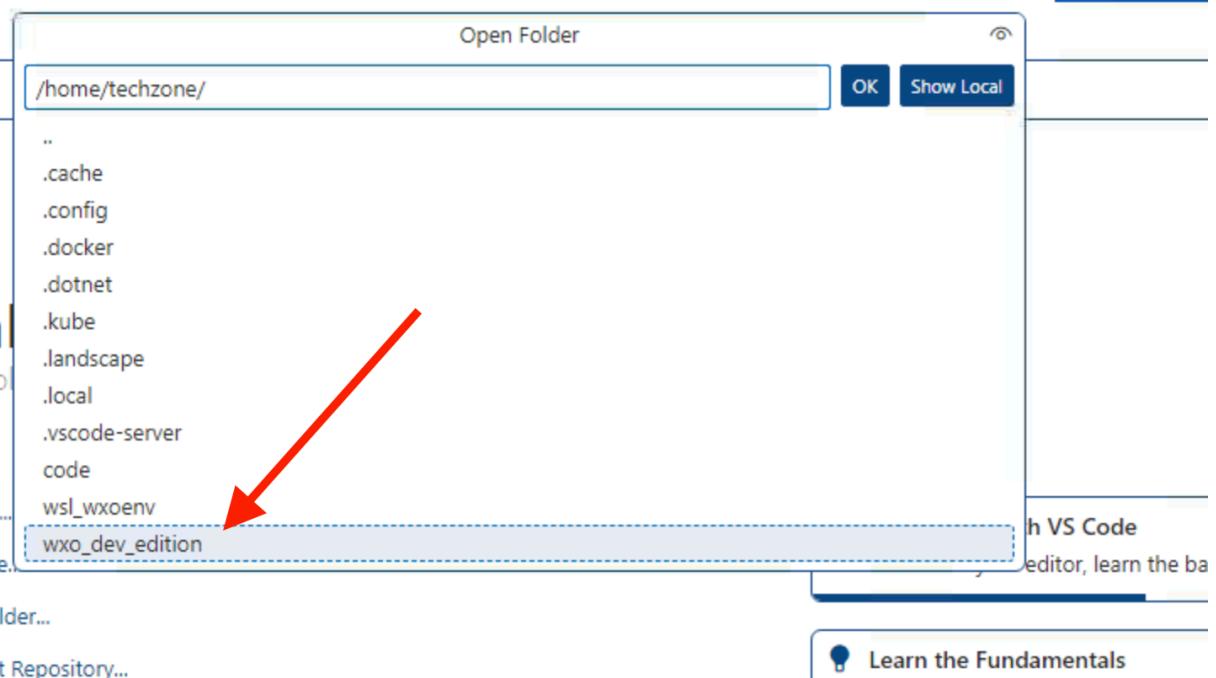
VS Code is already installed on the Windows-based virtual machine. To open it, simply type "code" into the search field at the bottom left of the screen. The "VS Code" app will automatically be offered as a choice, and you can open it by clicking on the app icon.



Once the application is open, select the File -> Open folder menu option.



Select the folder named `wxo_dev_edition` from the list.



Click OK. This will open the folder with all the files you are going to need during this lab.

Environments

To run the lab end to end, you need a number of environments.

watsonx.ai

For the lab, as well as for the install of the Developer's Edition of watsonx Orchestrate, you will need access to an IBM [watsonx.ai](#) Runtime instance, and specifically, a deployment space ID for that instance as well as an API key for the IBM Cloud account the instance is running in.

Your instructor should have given you access to the instances of watsonx Orchestrate and [watsonx.ai](#) that you will use throughout the bootcamp. To access them, you start out by logging into your IBM Cloud account at <https://cloud.ibm.com>. You can find the resources that you have access to in that account when going to the so-called "hamburger menu" on the top left of the page and clicking on Resource list .

The screenshot shows the IBM Cloud dashboard with a red arrow pointing to the 'Resource list' link in the left sidebar. The sidebar also lists other categories like Dashboard, Projects, Containers, Databases, Infrastructure, Observability, Platform Automation, Security, API Management, Cloud Pak for Data, Partner Center, SAP, Satellite, VMware, Watsonx, and Navigation settings. The main area displays several cards: 'Track emissions with Carbon Calculator' (Recommended, 1 min), 'Use Watson Assistant' (Popular, 2 min), 'Use Watson Studio' (Popular, 2 min), 'Build with Watson' (Popular, 3 min), and 'Retrieval Augmented Generation (RAG) Pattern' (Recommended). Below these are sections for Recent support cases, Planned maintenance (with 7 upcoming events), and Total emissions.

On the page with all your resources, you can find your [watsonx.ai](#) Runtime instance in the AI / Machine Learning section. The instance will have `watsonx Runtime` in the Product column. Click on the name of the instance.

The screenshot shows the 'Resource list' table with a red arrow pointing to the 'watsonx.ai Runtime' entry in the 'Product' column. The table has columns for Name, Group, Location, Product, Status, and Tag. It lists resources under three categories: Containers (2/139), Storage (1/1), and AI / Machine Learning (3/221). The AI / Machine Learning section includes entries for Watson Orchestrate-itz, wml-itz-wxo-684043ce0f81bb80d9c504, and ws-itz-wxo-684043ce0f81bb80d9c504.

Name	Group	Location	Product	Status	Tag
ce-itz-wxo-684043ce0f81bb80d9c504	itz-wxo-684043ce0f81bb80d9c504	Dallas (us-south)	Code Engine	Active	
cr-itz-qn1x2a5b	itz-wxo-684043ce0f81bb80d9c504	Dallas (us-south)	Container Registry	—	
cos-itz-wxo-684043ce0f81bb80d9c504	itz-wxo-684043ce0f81bb80d9c504	Global	Cloud Object Storage	Active	
Watson Orchestrate-itz	itz-wxo-684043ce0f81bb80d9c504	Dallas (us-south)	watsonx Orchestrate	Active	
wml-itz-wxo-684043ce0f81bb80d9c504	itz-wxo-684043ce0f81bb80d9c504	Dallas (us-south)	watsonx.ai Runtime	Active	
ws-itz-wxo-684043ce0f81bb80d9c504	itz-wxo-684043ce0f81bb80d9c504	Dallas (us-south)	watsonx.ai Studio	Active	

This will open the details page for the resource. Expand the Launch in drop-down list and click on IBM watsonx .

[Resource list](#) /wml-itz-wxo-684043ce0f81bb80d9c504 ✓ [Add tags](#) ↗

Manage

Plan



watsonx.ai Runtime in Cloud Pak for Data and watsonx

Put AI models to work. Deploy, monitor, and update models to gain insights on either platform. Work with foundation models on watsonx as a Service.

Launch in

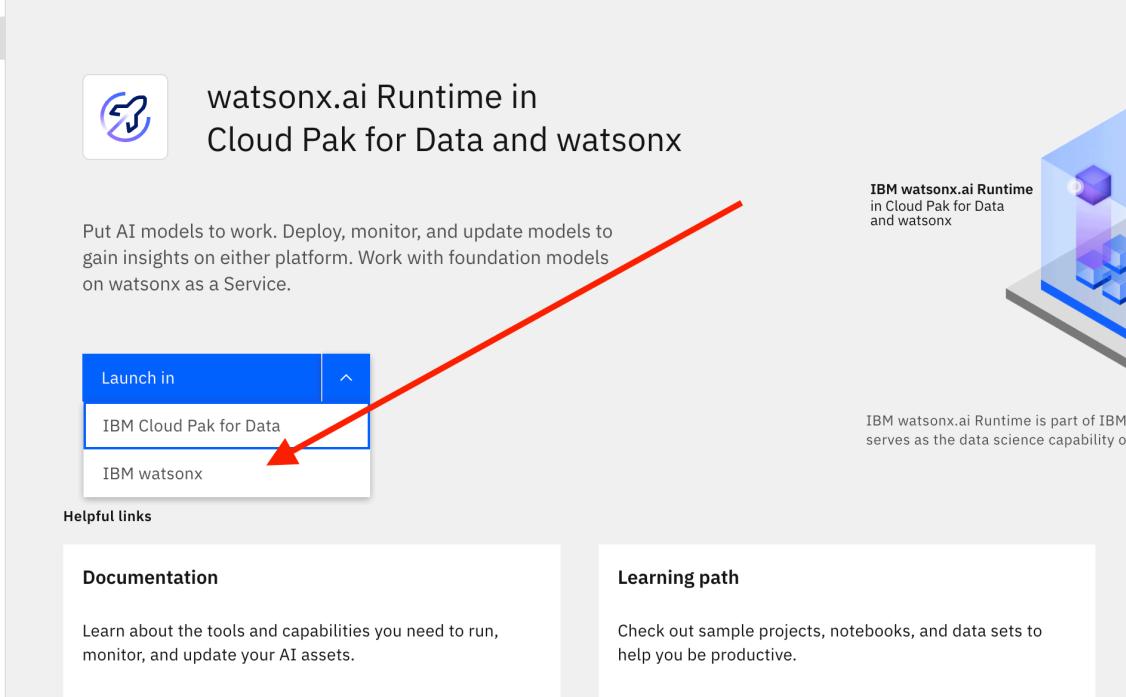
- IBM Cloud Pak for Data
- IBM watsonx**

Helpful links

Documentation
Learn about the tools and capabilities you need to run, monitor, and update your AI assets.
[Cloud Pak for Data →](#)

Learning path
Check out sample projects, notebooks, and data sets to help you be productive.
[Cloud Pak for Data →](#)

IBM watsonx.ai Runtime in Cloud Pak for Data and watsonx
IBM watsonx.ai Runtime is part of IBM Cloud Pak for Data and serves as the data science capability of Watson.



After opening the watsonx console, you can close both the Welcome and the Dive deeper pop-up windows. Now, click the 'hamburger menu' on this page, again at the top left of the page, and select View all deployment spaces .

The screenshot shows the IBM WatsonX interface. On the left, there is a dark sidebar with various navigation options: Home, Data, Projects, AI governance, Deployment spaces (which is currently selected and highlighted in blue), Resource hub, Administration, and Support. A red arrow points from the text "Click on the New deployment space button." to the "View all deployment spaces" link under the Deployment spaces section. To the right of the sidebar, there is a main content area titled "Open in: andre-events". It features a "Start chatting..." button and a "Open Prompt Lab" button. Below this, there is a "Developer access" section with fields for "Project or space" (set to "Project or space") and "Project ID" (set to "00000000-0000-0000-0000-000000000000"). There is also a "watsonx.ai URL" field containing "https://us-south.ml.cloud.ibm.com". A note below the URL says "Used to call watsonx.ai APIs such as LLM inferencing, embedding, training, and chatting."

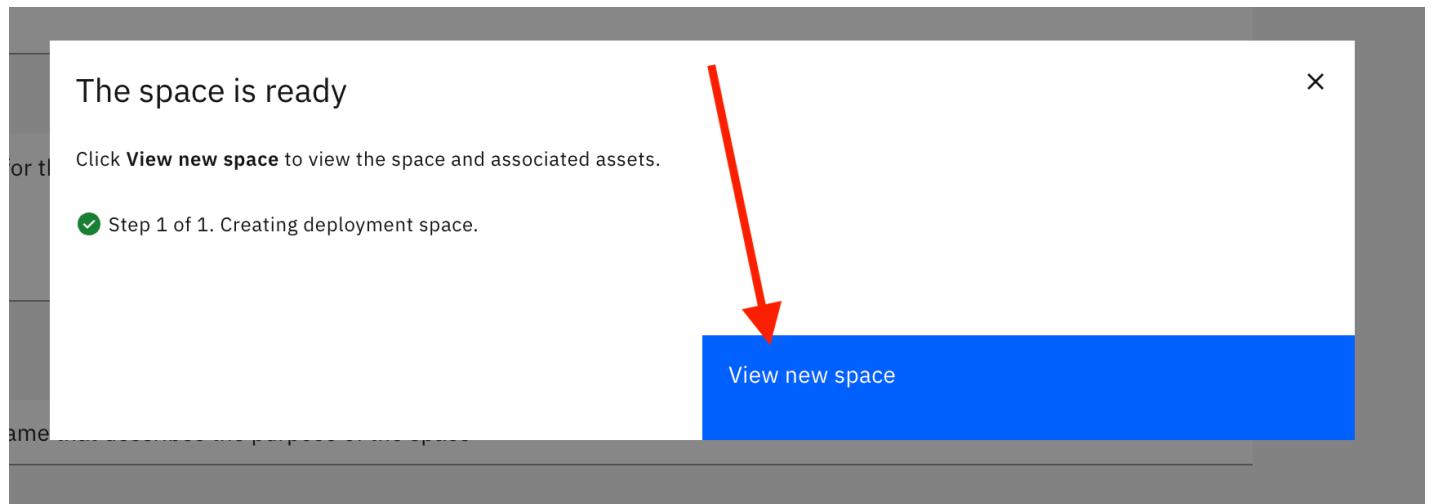
In the following view, depending on whether you have already run a different lab, you may or may not see any deployment spaces listed. However, here we will just create a new one. Click on the **New deployment space** button.

The screenshot shows a modal dialog for creating a new deployment space. At the top, there are icons for help, notifications, and account information, followed by the project name "2709027 - watsonx-events" and location "Dallas". On the right side of the header are buttons for "AT" and a three-dot menu. The main area of the dialog has a large blue button with white text that reads "New deployment space +". A red arrow points from the text "Click on the New deployment space button." to this blue button.

Give the new space a descriptive name. All other fields are optional. The **Storage** field should already be filled in. Click on **Create**.

The screenshot shows the 'Create a deployment space' form. The 'Name' field contains 'wxo ADK space'. The 'Description (Optional)' field has the text 'Deployment used for the watsonx Orchestrate ADK.' The 'Deployment stage' dropdown is set to 'Select or enter a name that describes the purpose of the space'. The 'Tags (optional)' section shows a dropdown menu with 'Find or create tags'. The 'Storage' section shows 'cos-itz-wxo-684043ce0f81bb80d9c504'. At the bottom right, there are 'Cancel' and 'Create' buttons, with the 'Create' button highlighted by a red arrow.

Once the space is ready, click on `View new space`.



On the details page for the new space, select the `Manage` tab.

wxo ADK space

Overview Assets Deployments Jobs **Manage**

Jump back in

Assets that you promote or add to the space show here.

View all

Deployments All ▾

Deployed	Failed
0	0

[View deployments](#)

Job runs

Active	Failed last 24 hours
0	0

[View jobs](#)

AI governance

Space history

① No notifications You will see your most recent notifications here.

On the Manage page, make sure the space is associated with your [watsonx.ai](#) Runtime instance, and set it if it is not. (Hit Save if you need to set it.)

wxo ADK space

Overview Assets Deployments Jobs **Manage**

Space

- General** (selected)
- Access control
- Environments
- Resource usage

General

Details

Name	wxo ADK space
Description	Deployment used for the watsonx Orchestrate ADK.
Space GUID	28c570ce-9a52-4de0-af4b-538c3451b6cc
Date created	Jun 6, 2025, 4:03 PM by Andre Tost (You)
Stage	Not provided
Tags	No tags are set to this space.
Last updated	Jun 6, 2025, 4:03 PM
Stage type	Pre-production

Storage

Storage used	0 Bytes
Name	cos-itz-wxo-684043ce0f81bb80d9c504
Bucket	a3b9b4a9-11c1-43e8-a345-0fb912ae553

[Manage in IBM Cloud](#)

watsonx.ai Runtime service

wml-itz-wxo-684043ce0f81bb80d9c504

① Cancel Save

The last step here is that we need to capture the Space GUID. You can find the GUID also on the Manage tab. Simply click on the icon next to it to copy it to your clipboard.

wxo ADK space

The screenshot shows the 'General' tab of a space named 'wxo ADK space'. The 'Space GUID' field contains the value '28c570ce-9a52-4de0-af4b-538c3451b6cc'. To the right of this field is a small blue clipboard icon with a white 'C' inside, which is highlighted by a red arrow pointing from the left.

General	
Details	Storage
Name wxo ADK space	Storage used 0 Bytes
Description Deployment used for the WatsonX Orchestrate ADK.	Name cos-itz-wxo-684043ce0f81bb80d9c504
Space GUID 28c570ce-9a52-4de0-af4b-538c3451b6cc	Bucket a3b9b4a9-11c1-43e8-a345-0fb912ae553
Date created Jun 6, 2025, 4:03 PM by Andre Tost (You)	Last updated Jun 6, 2025, 4:03 PM
Stage <i>Not provided</i>	Stage type Pre-production
Tags No tags are set to this space.	Manage in IBM Cloud
watsonx.ai Runtime service	
wml-itz-wxo-684043ce0f81bb80d9c504	

The .env file

The Space GUID, as well as a number of other environment variables, goes into a file called `.env`. This file should exist in **the root folder** of where you extracted the content of the repo that was provided to you by your instructor (this file is also in that repo, of course), in other words, it should be at the same level as the `usecases` or `environment-setup` subfolders.

- If you are using the virtual machine with a pre-installed ADK, you already have this file in the `wxo_dev_edition` folder in the WSL environment within that virtual machine.
- If you are running this on your local machine, you should have already unzipped the file with materials into a folder of your choosing, as described [above](#). Create an empty `.env` file and make sure you place the `.env` file in that same folder.

You can edit the file with an editor of your choosing. We recommend using VS Code for this.

You can create and edit this file with an editor of your choosing, or simply run the following command on the command line:

```
echo 'WATSONX_SPACE_ID=[paste the Space GUID from your clipboard here]' >> .env
```

API key

You also need an API key for the IBM Cloud account that your `watsonx.ai` instance is on. To obtain one, open the IBM Cloud console (<https://cloud.ibm.com>) as before and select `Access (IAM)` option from the `Manage` dropdown at the top of the page.

The screenshot shows the IBM Cloud dashboard with the 'Manage' menu open. The 'Access (IAM)' option is highlighted with a red arrow. Other options in the menu include Account, Billing and usage, Catalogs, Enterprise, Security and access, and Context-based restrictions.

On the following page, select API keys from the menu on the left.

The screenshot shows the IBM Cloud IAM Overview page. The left sidebar has a menu with several items: Overview (selected), Dashboard, Manage identities (with sub-options: Users, Trusted profiles, Service IDs, API keys), Identity providers, Manage access (with sub-options: Access groups, Authorizations, Roles), Gain insight, Settings, Documentation, and Enterprise IAM Docs. A red arrow points to the 'API keys' item under 'Manage identities'. The main content area features a large title 'IBM Cloud Identity and Access Management' and a subtitle 'Securely authenticate users for platform services and control access to resources.' Below this are sections for 'Invite users', 'Create access group', and 'Help'. A 'What's new?' section is present, with a recent update about resource attribute-based conditions. The bottom section discusses centrally administering multi-account environments.

You may or may not already have an API key listed, and if so, feel free to use it (note that you cannot see the value of an existing API key after you initially created it). Or you can simply create a new one by clicking on the Create button.

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.

Unused or overly permissive API keys increase the risk of unauthorized access. Regularly review the [Inactive identities report](#), rotate keys, and apply only the minimum required permissions.

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.

The screenshot shows a table listing API keys. The columns are: Status, Name, Description, Date created, and Enabled. There are two entries:

- andre-events-key: Enabled, created on 5-19-2025 15:21 GMT.
- cpd-apakey-IBMid-3100020N5U-2025-06-04T14:02:35Z: Enabled, created on 6-4-2025 14:02 GMT. Description: API key created/managed by task credentials. It is managed for your use with Watson Studio operations. Please do not delete here.

At the top right of the table is a blue 'Create' button with a red arrow pointing to it. Below the table are pagination controls: 'Items per page: 25' and 'Page 1'.

Give the new key a descriptive name, and click on Create .

Create IBM Cloud API key

Name: APIKeyforwxoADK

Description (optional): Enter description

Leaked action

If API key is discovered to have been leaked out in the world, what would you like the system to do?

Disable the leaked key
 Delete the leaked key
 Nothing

Session management

Enable session management for CLI logins? [\(i\)](#)

Yes No

Create

Once the key has been successfully created, make sure you copy its value to the clipboard by clicking on the Copy link. As mentioned above, you won't be able to retrieve this value later.

P1 Keys

ate, 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 platforms 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](#).

The screenshot shows a modal dialog box with the title "API key successfully created". The message inside says: "Copy the API key or click download to save it. You won't be able to see this API key again, so you can't retrieve it later. The API key is no longer displayed after 291 seconds." Below this, there is a table titled "API key" with one row visible. The row contains a redacted API key value, a copy icon, a download icon, and a timestamp of "6-4-2025 14:52 GMT". At the bottom of the table, there is a note: "API key created/managed by task credentials. It is managed for your use with Watson Studio operations. Please do not delete here." A red arrow points from the "Copy" button in the modal to the "Copy" icon in the table row.

API key	Created/Managed By	Created On
.....	API key created by task credentials. It is managed for your use with Watson Studio operations. Please do not delete here.	6-4-2025 14:02 GMT

The API key also goes into the .env file, and you can add it via editor or command line:

```
echo 'WATSONX_APIKEY=[paste the API key from your clipboard here]' >> .env
```

Entitlement key

Below, you will install the watsonx Orchestrate Developer Edition, which consists of a number of container images that are downloaded from the IBM registry during install. To authenticate with this registry, you need an "entitlement key". Your instructor will provide this key for you.

You can add the key to your .env file via editor or by running the following on the command line:

```
echo 'WO_ENTITLEMENT_KEY=[add the entitlement key you received from your instructor here]' >> .env
```

watsonx Orchestrate ADK

As mentioned above, the ADK allows hosting the core Orchestrate components on a developer laptop. For the lab, you can choose if you want to run the ADK on your own laptop or on a virtual machine that will be provided to you by your instructor.

Local machine

To run it on your own laptop, you need to install

- [Docker](#) or [Rancher](#)

- the containers that run as part of the ADK will require ~12GB of memory, so you need to allocate at least that much to the virtual machine hosting the container runtime
- Python 3.11
- Visual Studio Code

Once you have these prerequisites available, you can install the ADK by following the instructions at [the ADK install page](#).

Note: These instructions were created for a specific version of the ADK, namely version **1.5.1**. We recommend you specify that version when running the install: `pip install ibm-watsonx-orchestrate==1.5.1`.

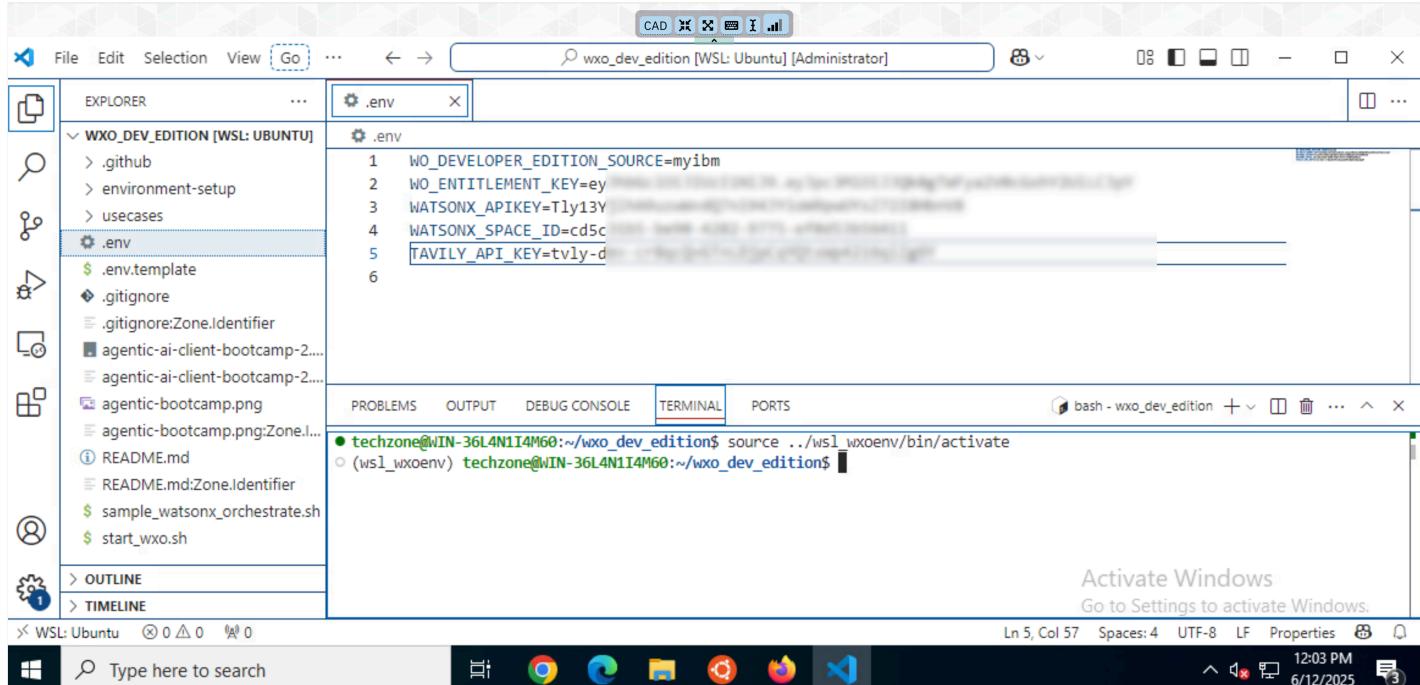
You also need to install the watsonx Orchestrate Developer Edition, which is part of the ADK, by following the related [install instructions](#). However, **DO NOT** set up the .env file as described in the instructions! You already have the right values in your .env file if you followed the instructions above.

After you created the .env file with the values given to you, you can follow the instructions to start the server for the first time as documented [here](#). Note that the first time you run it, it will download all the required container images from the IBM image registry, which will take some time.

Virtual machine

At this point, you should have filled in the required values into the .env file using VS Code as described above. You should also have a command line terminal open in the UI, with `wxo_dev_edition` as the current folder.

Alternatively, you can also use the built-in terminal window in VS Code, which is located below the main editor window. Make sure you activate the Python environment as shown in the picture below.

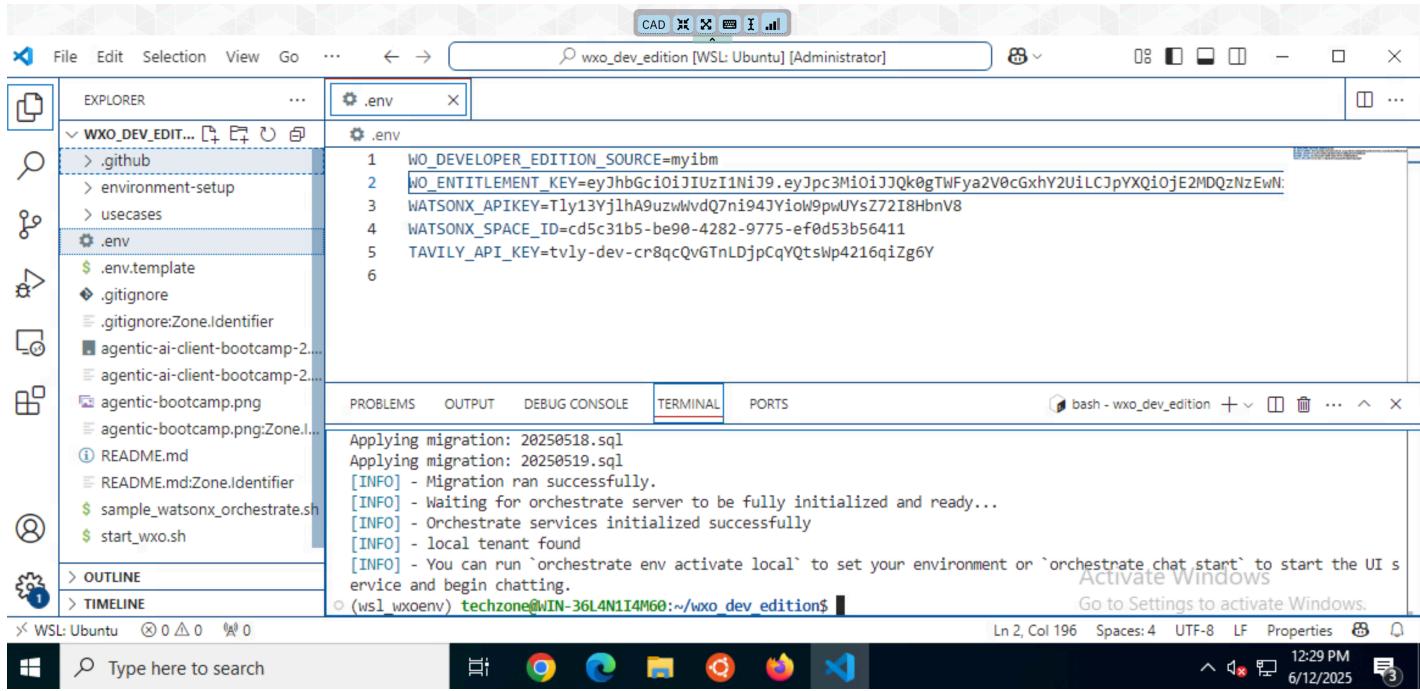


You don't need to set the Tavily API key here, we will show how to obtain and set that key [below](#).

You can start the Orchestrate server by entering the following command:

```
orchestrate server start --env-file .env
```

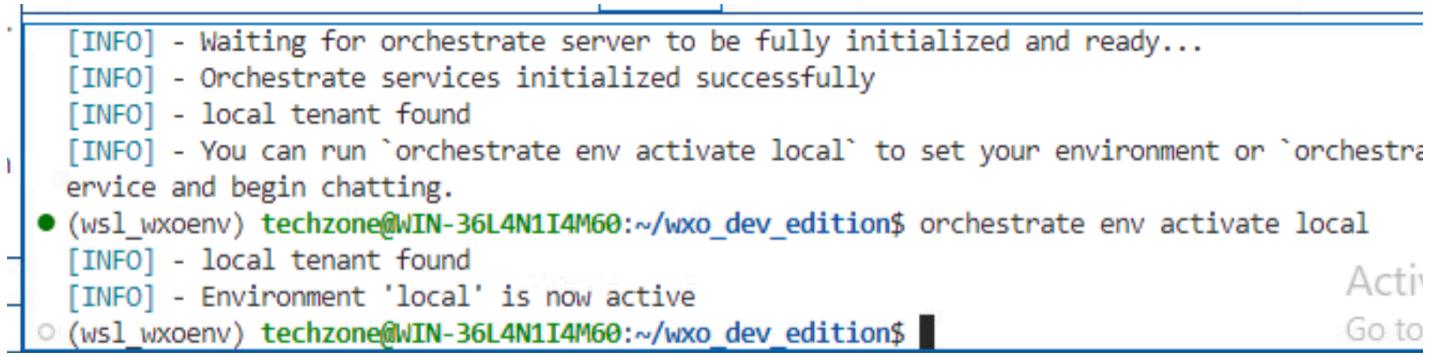
When running it for the first time, it may take a bit longer to start, depending on whether it has to download the latest versions of the container images (the images should all be cached in the virtual machine already, though).



The screenshot shows a Windows Start Menu with the 'orchestrate' application pinned to the left. The pinned tile has a blue background with white text. Below the pinned tiles is a search bar with the placeholder 'Type here to search'. To the right of the search bar is a taskbar with several icons: File Explorer, Edge browser, File Explorer again, Task View, and a terminal window icon.

Once you see the message that the server has been started and that you should activate the local environment, enter the following:

```
orchestrate env activate local
```



The terminal window shows the following output:

```
[INFO] - Waiting for orchestrate server to be fully initialized and ready...
[INFO] - Orchestrate services initialized successfully
[INFO] - local tenant found
[INFO] - You can run `orchestrate env activate local` to set your environment or `orchestrate chat start` to start the UI service and begin chatting.
● (wsl_wxoenv) techzone@WIN-36L4N1I4M60:~/wxo_dev_edition$ orchestrate env activate local
[INFO] - local tenant found
[INFO] - Environment 'local' is now active
○ (wsl_wxoenv) techzone@WIN-36L4N1I4M60:~/wxo_dev_edition$
```

watsonx Orchestrate

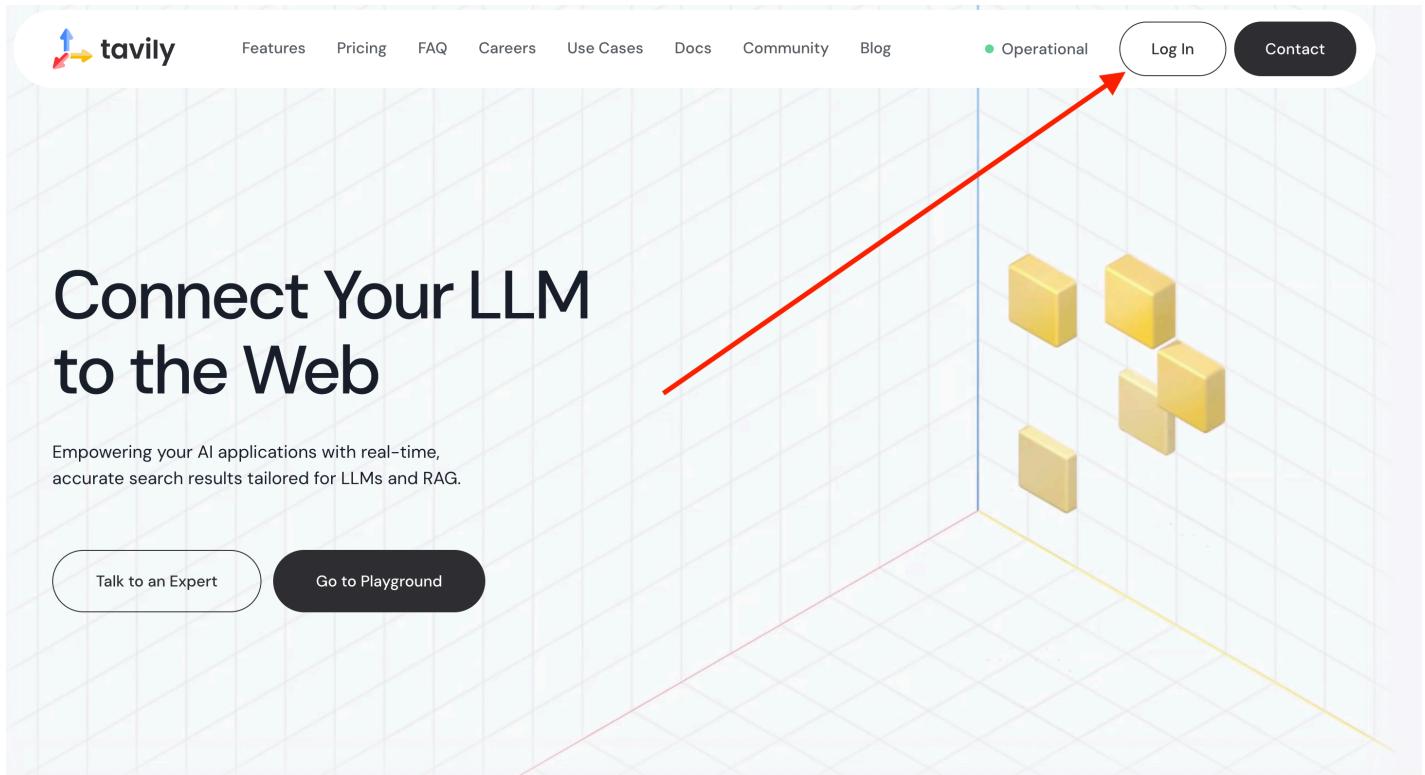
In this lab, you will create a number of components (tools, agents, etc) in your local environment and run and test them there, without the need to access an instance of watsonx Orchestrate in the cloud. However, the last part of the lab describes how you can take the same components and easily deploy and run them on a watsonx Orchestrate SaaS instance. You need such an instance for that part of the lab.

Your instructor will provision both the [watsonx.ai](#) and the watsonx Orchestrate instances for you and you can find the watsonx Orchestrate resource in the IBM Cloud resource list. This is only needed for the last part of the lab.

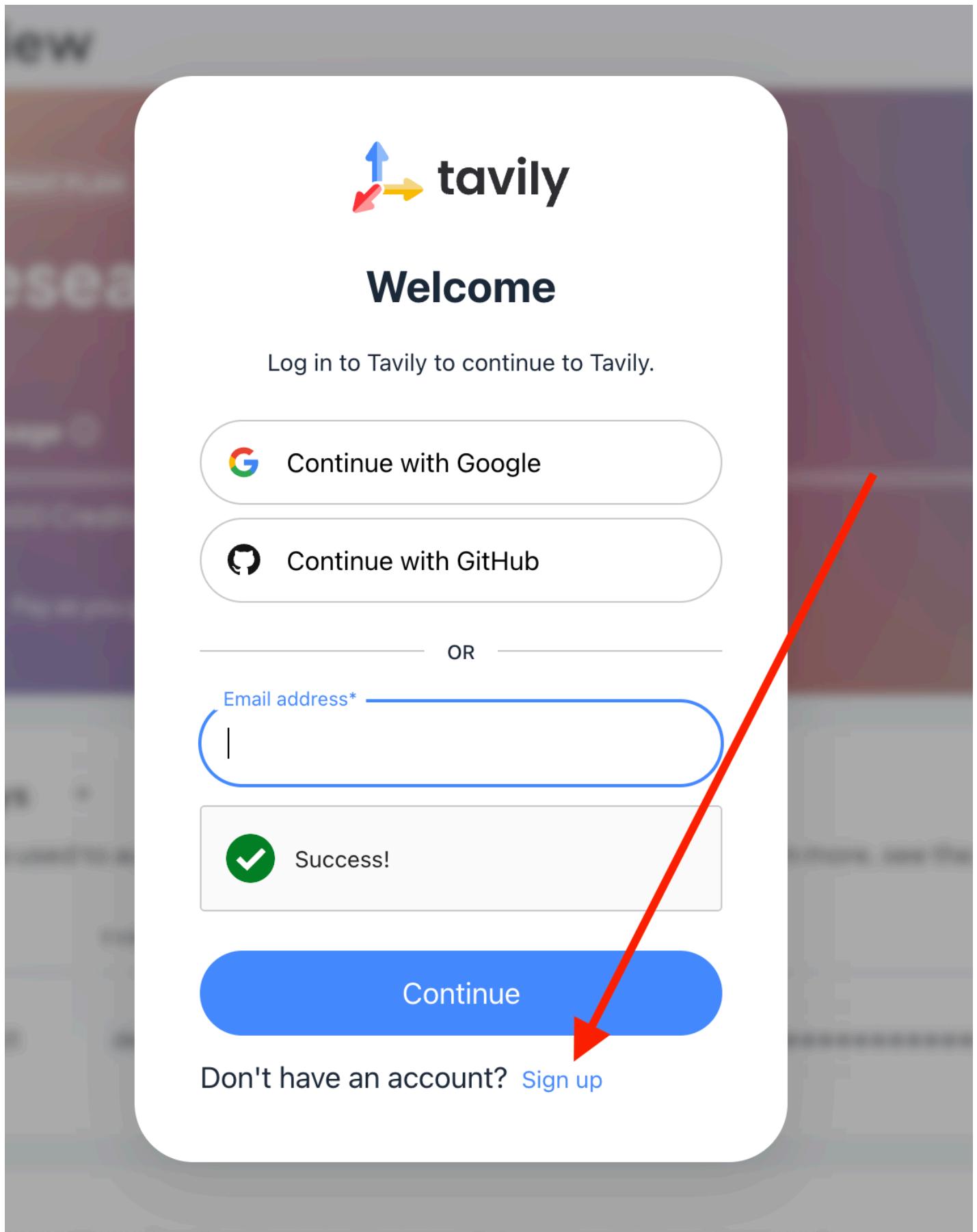
Tavily

One of the tools you will create during the lab is a web search tool that takes advantage of a service called "Tavily". To use it, you need an API key that is passed with every search request.

Go to <https://www.tavily.com/> and click on Log in .



Now click on Sign up .



You can sign up with your Google ID, or Github ID, or your email address. Once you have successfully completed the signup process and can log into the service, your page should look like this:

The screenshot shows the Tavily Overview page. On the left is a sidebar with links like Overview, API Playground, Use Cases, Billing, Settings, Documentation, and Tavily MCP. The main area has a 'CURRENT PLAN' section titled 'Researcher'. Below it is an 'API Usage' section with a progress bar and a 'Pay as you go' toggle. At the bottom is a table for 'API Keys' with one row: 'No API keys found. Click the "+" button above to create a new key.' A red arrow points to the '+' button in the 'API Keys' section.

Click on the Plus sign as shown in the image above. Name your key "default". After it has been created, you can copy its value to the clipboard by clicking on the copy icon next to your key:

The screenshot shows the Tavily Overview page after creating an API key. The 'API Keys' table now has one row: 'default' (Type: dev, Usage: 0%, Key: tvly-dev-*****). A red arrow points to the copy icon (a clipboard icon) next to the key value 'tvly-dev-*****'.

To complete the setup for this use case, we will add the Tavily API key to the .env file as before, either with an editor of your choice or via command line:

```
echo 'TAVILY_API_KEY=[copy the Tavily API key from your clipboard here]' >> .env
```

This is it! You are now ready to proceed to the [detailed lab instructions](#).