

[Event dashboard](#) > [Prerequisites](#) > At an AWS Event (Setup)

At an AWS Event (Setup)

Sign in a Workshop Studio AWS Account

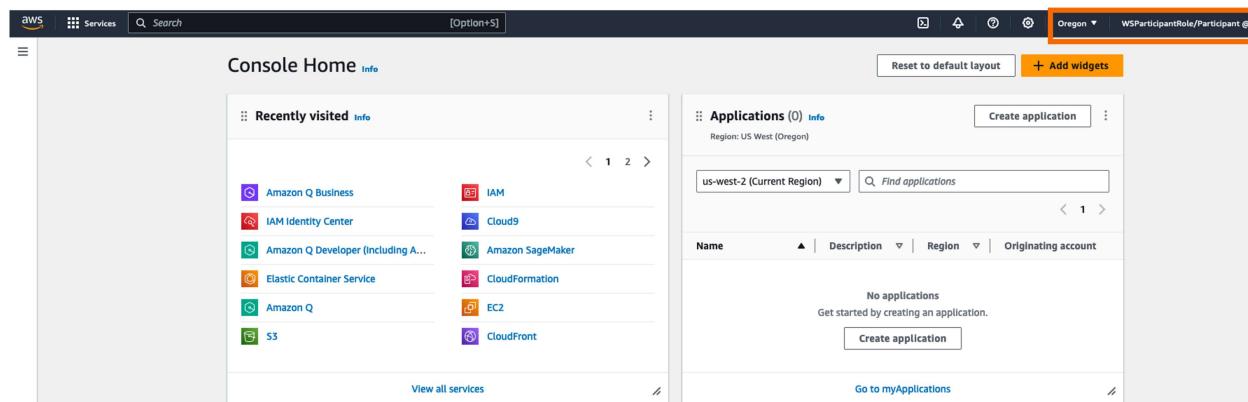
If you participate in an AWS Immersion Day, public workshop, or a similar instructor-led event you get access to a temporary AWS Account already pre-configured with a Visual Studio Code IDE and all required resources.

Before you start

- Log out from all AWS accounts from all browser windows
- Review the terms and conditions of the event. Do not upload any personal or confidential information in the account
- The AWS account is only available for the duration of this workshop and you will not be able to retain access after the workshop is complete. Backup any materials you wish to keep access to after the workshop
- Workshop deploys all pre-provisioned infrastructure to a specific region. Check what AWS Region is used

Workshop Studio AWS Account access

1. Go to [AWS Workshop Studio](#) and click **Get started**:
2. Enter the 12 digit event access code distributed by the instructor and click **Next**.
3. choose **Email one-time password (OTP)** to sign in.
4. Enter your email, receive a one-time email passcode (9-digit), and sign in with it.
5. Review the terms and conditions associated with this event, and then click **Join event**.
6. After joining the event, you see the page with event information and workshop details. Choose **Open AWS console**.
7. Once logged in you should see your User as WSParticipantRole/Participant in the top right corner.



Remote Desktop Setup

Step 1: Enable RDP access to your instance

Before you can connect to your instance, you need to configure the security group to allow RDP access from your IP address:

1. Navigate to the [EC2 Security Groups console](#)
2. Find and click the security group that has `UbuntuDesktopSecurityGroup` in its name
3. Click the **Edit inbound rules** button
4. Find the RDP rule (port 3389) and change the **Source** from `0.0.0.0/32` to **My IP**
5. Click **Save rules**

The screenshot shows the 'Edit inbound rules' interface for a security group. A single rule is listed: 'RDP' (Type), 'TCP' (Protocol), port '3389' (Port range), 'My IP' (Source). The 'Description' field contains 'RDP access from My IP'. At the bottom right, there are 'Cancel', 'Preview changes', and 'Save rules' buttons, with 'Save rules' being the active one.

Security Note: This step restricts RDP access to only your current IP address, which is a security best practice. If you change networks or your IP changes, you may need to repeat this step.

Step 2: Set up the Remote Desktop client

For Windows:

- Use the native **Remote Desktop Connection** client that's already installed on Windows (no additional installation needed)
- You can find it by searching for "Remote Desktop Connection" in the Start menu

For Mac:

- Download and install the [Windows App from Mac App Store](#)

Step 3: Connect to your instance

First, get your connection details:

1. Navigate to the [Workshop Studio Event Outputs](#).
2. Scroll down until you see the **Event Outputs** section at the bottom of the page.
3. Note the following information:
 - **RDPAddress**: The address you'll use to connect to your instance
 - **RDPUsername**: The username for logging in
 - **RDP>Password**: The password for logging in (copy this for later use)

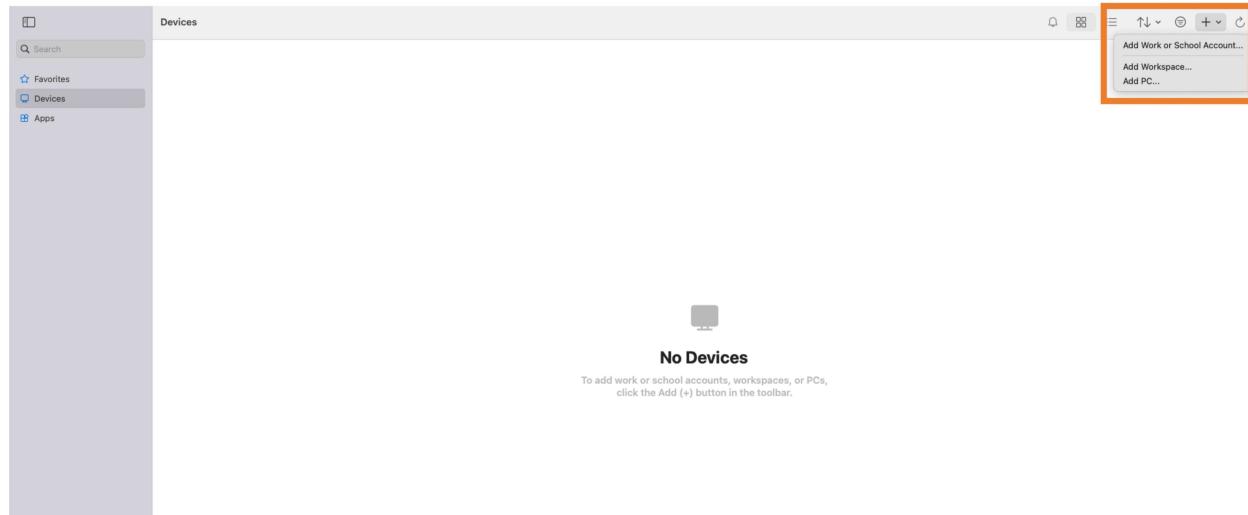
Then connect to your instance:

For Windows:

1. Open Remote Desktop Connection (search for it in the Start menu).
2. In the "Computer" field, enter the **RDPAddress** value you noted earlier.
3. Click "Connect".
4. When prompted, enter the **RDPUsername** and **RDP>Password** values you noted earlier, then click "OK".

For Mac:

1. Open the Microsoft Remote Desktop app.
2. Click the plus (+) icon in the top right corner, then select "Add PC".
3. For the PC name, enter the value of **RDPAddress** that you noted earlier, then click "Add".



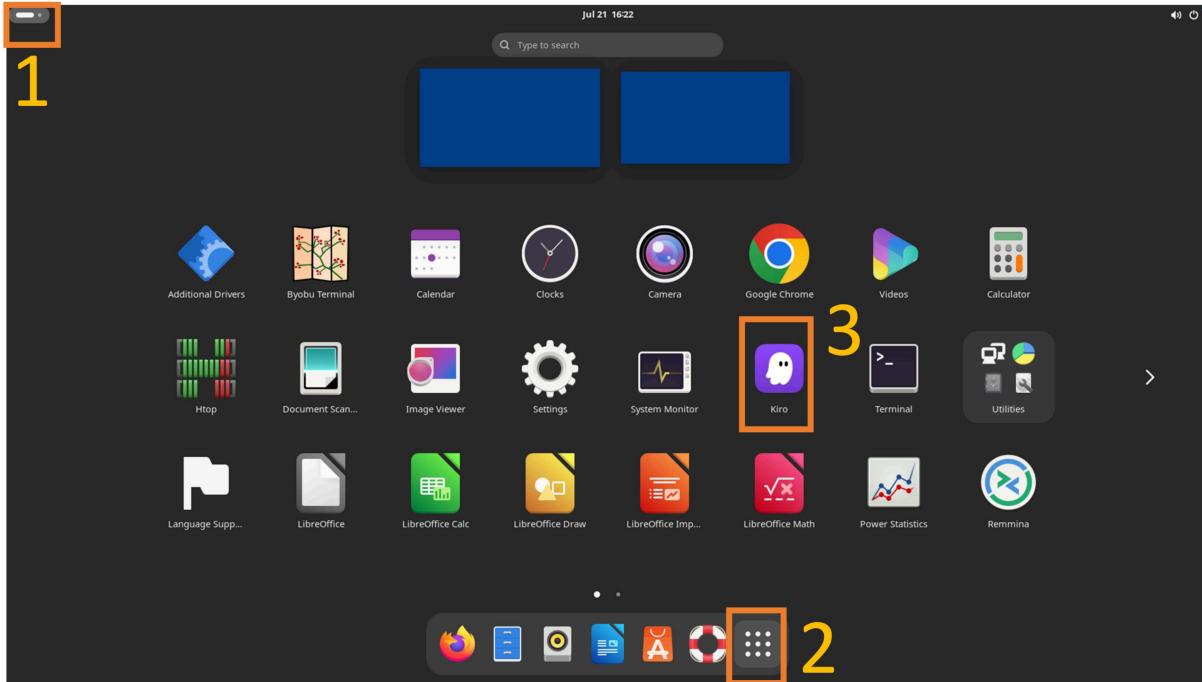
4. Back at the home screen, double-click your newly added instance to bring up the login window.
5. Enter the **RDPUsername** and **RDPPassword** values you noted earlier, then click "Connect".

Getting Started with Kiro

Open Kiro

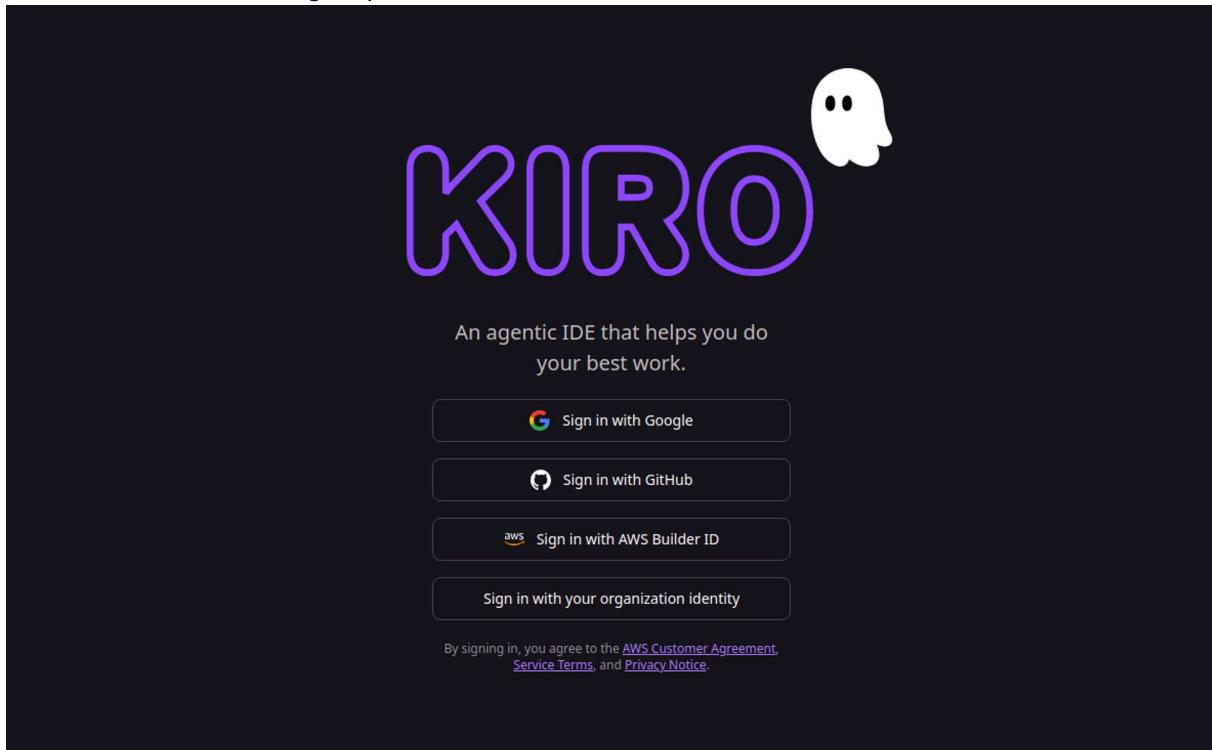
Once logged in to the Ubuntu Desktop:

1. You can configure keyboard settings if needed.
2. To open Kiro, click the menu icon on the top left of the screen.
3. Click the "All Apps" button at the bottom of the menu.
4. Find and click the Kiro icon to launch the application.

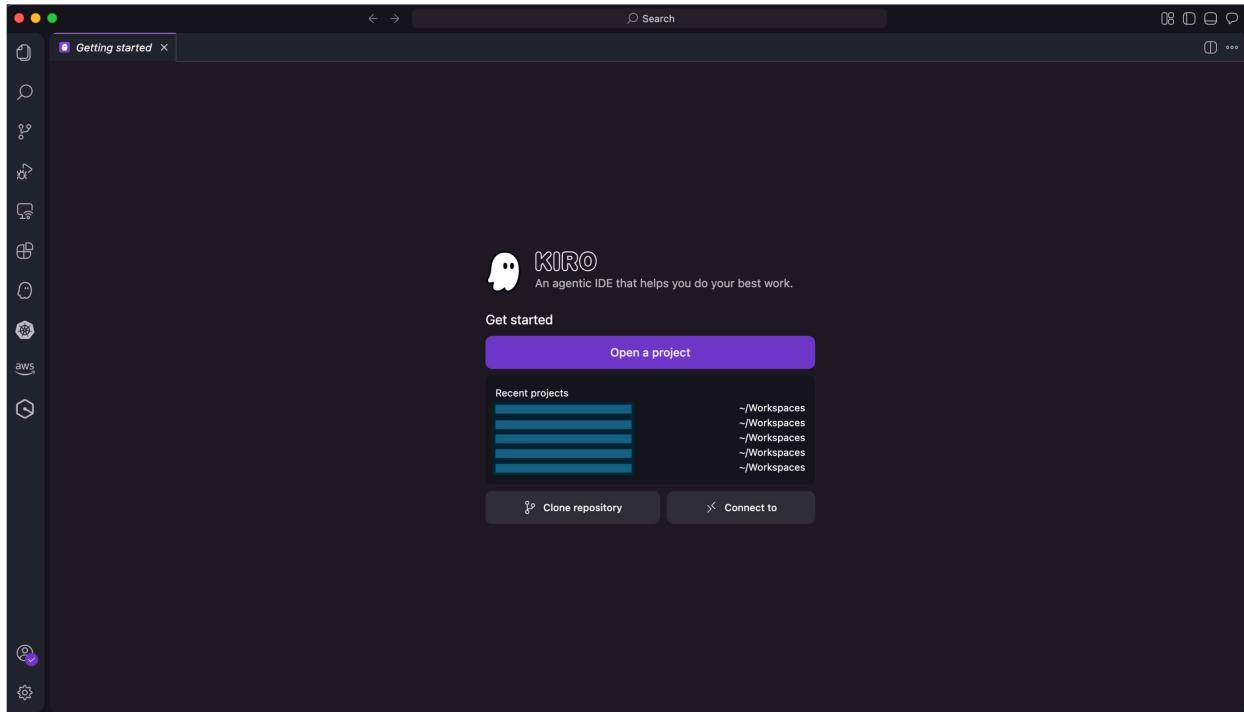


Sign in

- When you open Kiro for the first time you will be asked to login with a provider of your choice that include social and AWS login options. Learn more about the [auth methods](#).



- Once logged in, you can choose to import your VS Code settings and extensions. You can skip this step. Select your preferred theme from the available options, then allow Kiro to set up shell integration, enabling the agent to execute commands on your behalf.
- Finally, you'll arrive at the welcome page.



Open a Project

Create a folder with name **kiro-folder** as the project directory in this workshop.

Launch Kiro and open your project (i.e. **kiro-workshop**):

- Use File > Open Folder to select your project directory
- Or drag and drop your project folder into Kiro
- Or go to the command line and run `kiro .` from your project directory

Access the Kiro Panel

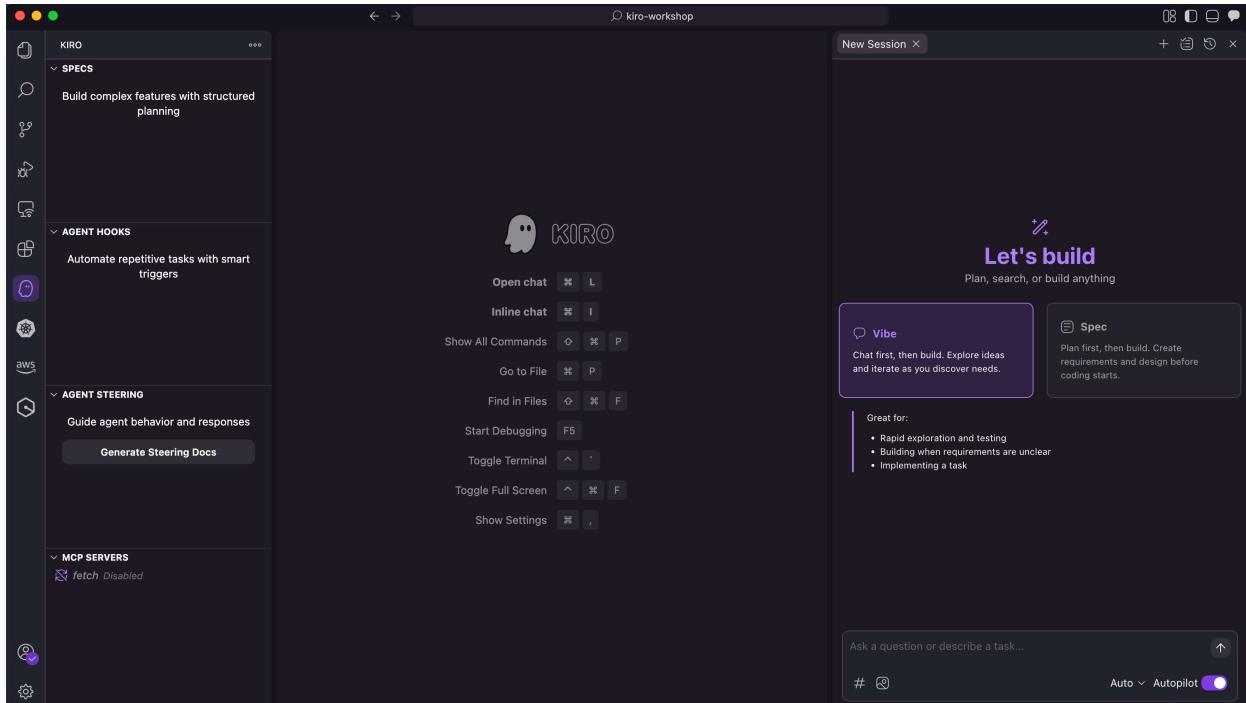
Click the Kiro Ghost icon in the activity bar (left sidebar) This panel provides access to all of Kiro's AI-powered features

Start a Chat Session

The chat pane should be open by default.

If not, there are multiple ways to access the chat in your development environment:

- Keyboard Shortcut: Press `Cmd+L` (Mac) or `Ctrl+L` (Windows/Linux) to open the chat panel
- Command Palette: Press `Cmd+Shift+P` (Mac) or `ctrl+Shift+P` (Windows/Linux) and search for "Kiro: Open Chat"
- Secondary Side Bar: Click the Kiro chat icon toggle using `Cmd+Opt+B` (Mac) or `Ctrl+Alt+B` in the top bar on the right to open the chat panel

[Previous](#)[Next](#)

pwhoae ▾