

## Building with Foundation Models on Amazon SageMaker Studio

Building with Foundation Models on Amazon SageMaker Studio

Introduction to Workshop Studio Setup

### SageMaker Spaces: JupyterLab and Code Editor

Lab 0 - Deploy Llama2 and Embedding Models

Lab 1 - Setup an LLM Playground on Studio

Lab 2 - Prompt Engineering with LLMs

Lab 3 - Retrieval Augmented Generation (RAG) using PySpark on EMR

▶ Lab 4 - Fine-Tune Gen AI Models on Studio

Lab 5 - Foundation Model Evaluation

### AWS account access

[Open AWS console \(us-east-1\)](#)

[Get AWS CLI credentials](#)

[Get EC2 SSH key](#)

[Exit event](#)

# SageMaker Spaces: JupyterLab and Code Editor

## Create Spaces

A Space is a contained provisioned ML Compute 'Space' that can either consist of,

1. [JupyterLab Environment](#)
2. [Code Editor Environment](#)

ⓘ We're going to create and run a Space of each type throughout this workshop.

### Create a JupyterLab Space

JupyterLab is an interactive development environment for working with notebooks, code, and data. It's an evolution of the Jupyter Notebook, offering a flexible and powerful user interface that supports a wide range of workflows in data science, scientific computing, and machine learning. Data scientists can easily combine code, data, and visualizations in a single, intuitive interface, streamlining the development of machine learning models. JupyterLab facilitates exploratory data analysis, visualization, and collaboration, making it easier to prototype, debug, and share results. Its support for various programming languages and integration with big data tools enhances its utility in developing complex machine learning workflows.

1. Click on JupyterLab Application from Studio UI Application Menu and then Create JupyterLab Space

The screenshot shows the SageMaker Studio interface. On the left, there's a navigation bar with icons for JupyterLab, Studio, Canvas, Code Editor, and Studio CLI. The JupyterLab icon is highlighted with a red box and an arrow points from it to the text "JupyterLab Application from Menu" located just below the navigation bar. The main area is titled "Home" and contains sections for "Overview" and "Getting started". The "Overview" section features a large orange "jupyter" logo and a call-to-action button "View JupyterLab spaces >".

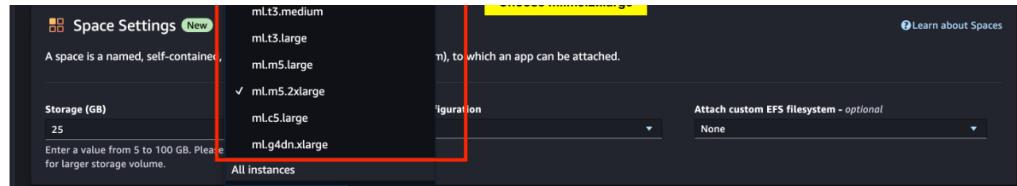
The screenshot shows the "JupyterLab" configuration page. At the top right, there are two buttons: "Create a New Space" and "+ Create JupyterLab space", both highlighted with red boxes. The main area has sections for "About" and "See features". Below that is a search bar and a filter for "Running" spaces. A table lists spaces with columns for Name, Application, Status, Type, Last modified, and Action. The "Action" column for the first space shows a "Run space" button.

2. Configure your Space with a Compute Type, Image Type and Storage Size (in GB). Click Run Space to kick start a JupyterLab Space. Give it 30-60 Seconds, a new Space should be ready for you to launch.

#### Important

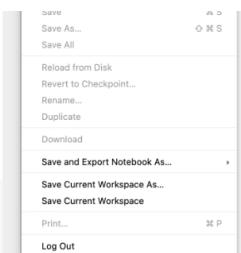
Please ensure you select ml.m5.2xlarge as the instance type and raise storage size to 25 GB.

The screenshot shows the "JupyterLab" configuration details page. It displays the space name "jlab4-space", status "Stopped", and instance type "ml.m5.2xlarge". The image type is set to "SageMaker Distribution 1.2". A yellow box highlights the "Choose ml.m5.2xlarge" button at the bottom of the instance type dropdown.



Give it 30-60 Seconds, a new Space should be ready for you to launch.

1. Clone the repository below. To clone a repo launch a terminal by navigating to [File > New > Terminal](#) and pasting the command below,



## Create a Code Editor Space

A web IDE (Integrated Development Environment) Code Editor is a browser-based platform that provides comprehensive tools for coding, debugging, and deploying software. Code Editor, based on Code-OSS, Visual Studio Code Open Source, enables you to write, test, debug and run your analytics and machine learning code. It is fully integrated with SageMaker Studio and supports IDE extensions available in the Open VSX Extension Registry. SageMaker Code Editor supports a variety of machine learning frameworks, and is widely used for Data Science and MLOps workflows. Its rich ecosystem of extensions, integrated Git support, and robust code editing and debugging features make it a favorite among developers for its efficiency and customizability.

1. Click on Code Editor Application from Studio UI Application Menu and then [Create Code Editor Space](#)

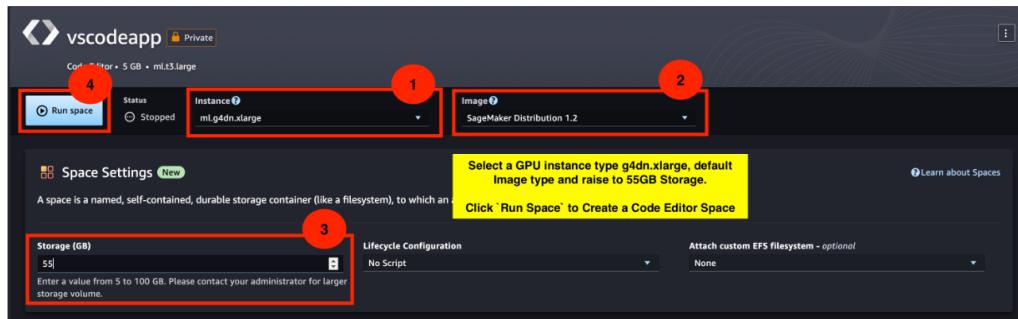
The screenshot shows the SageMaker Studio Home page. On the left, there's a sidebar with various application icons: JupyterLab (orange), RStudio (blue), Canvas (purple), and Code Editor (highlighted with a red box). A yellow callout box points to the 'Code Editor' icon with the text 'Code Editor Application from Menu'. The main area shows sections for 'Home', 'Overview', 'JupyterLab', 'Code Editor', 'Prebuilt and automated solutions', 'JumpStart', and 'AutoML'.

The screenshot shows the 'Code Editor' application page. At the top right, there's a 'Create a New Space' button with a plus sign, which is highlighted with a red box. Below it, there's an 'About' section with a warning icon and text about the Code Editor being based on Code-OSS, Visual Studio Code Open Source. There's also a search bar and a filter for 'Running' spaces.

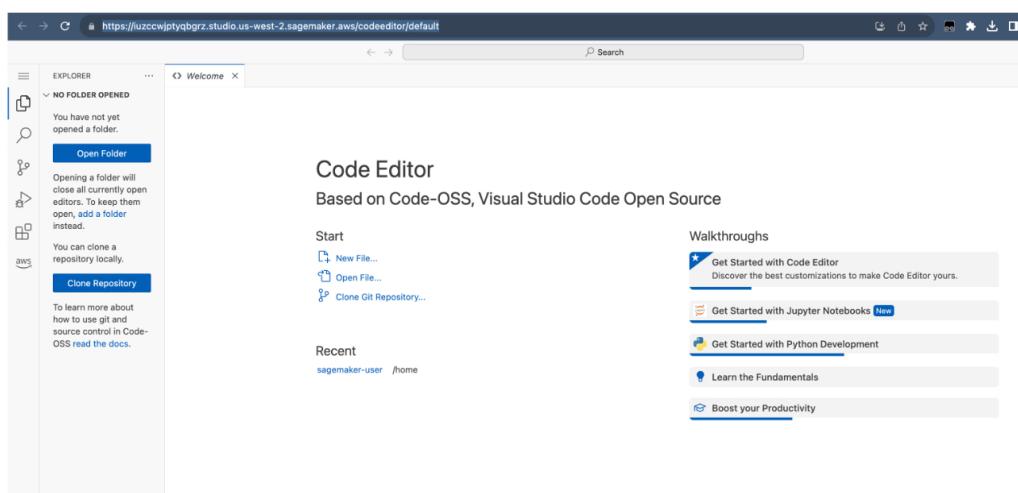
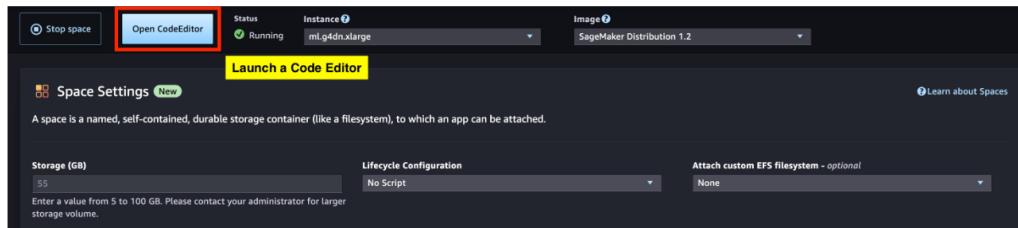
2. Similar to JupyterLab, for Code Editor, configure a Space with a Compute Type, Image Type and Storage Size (in GB). Click Run Space to kick start a Code Editor Space.

**Important**  
Please ensure you select `ml.g4dn.xlarge` as the instance type and raise storage size to 55 GB.

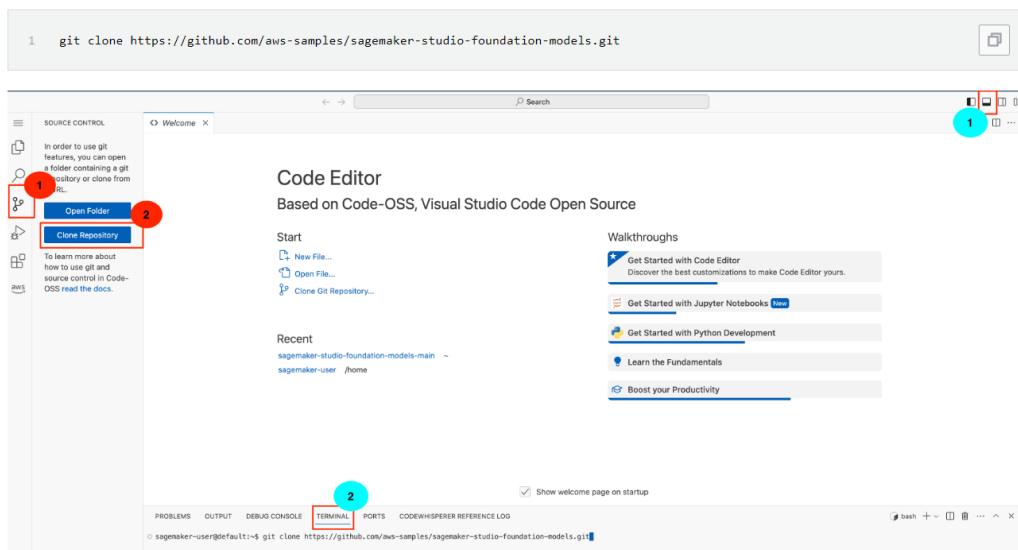
The screenshot shows the 'vscodeapp' configuration page. It has tabs for 'Run space' (selected) and 'Status' (Stopped). Under 'Space Settings', there's a note about naming the space. The 'Storage (GB)' field is set to 5. The 'Instance' dropdown is set to 'ml.g4dn.xlarge' and is highlighted with a red box. A yellow callout box points to this dropdown with the text 'Choose ml.g4dn.xlarge'. Other instance types listed include ml.t3.medium, ml.t3.large, ml.m5.large, ml.m5.2xlarge, ml.c5.large, and ml.g4dn.xlarge. There's also a 'Choose ml.g4dn.xlarge' button. At the bottom, there are buttons for 'Attach custom EFS filesystem - optional' (None selected) and 'Learn about Spaces'.



Give it 30-60 Seconds, a new Space should be ready for you to launch.



1. Clone the repository below. To clone a repo launch on Code Editor, you can use the Git GUI from left menu pane (red flow) OR launch a terminal by toggling the bottom pane and paste the command below (blue flow)



#### JupyterLab and/or Code Editor Setup Success!

If you're able to successfully validate the launch of JupyterLab and Code Editor, you're all set!

Let's move on to the labs portion of this Workshop. We have some exciting Labs awaiting for you!

