



Launching GenAI Tools on FASRC Clusters - Session 2

Nathan Weeks

Harvard - FAS Research Computing

Learning objectives

- How to set up the following interfaces to use approved AI model providers for coding:
 - JupyterLab (FASRC Open OnDemand)
 - Jupyter AI
 - RStudio Server (FASRC Open OnDemand)
 - Posit Assistant
 - VSCode
 - Codex, Claude Code for VS Code extensions
- Permissions modes for safer use on the FASRC cluster

Training Material

```
# Login to Cannon
ssh <username>@login.rc.fas.harvard.edu

# Check current location; change if desired:
> pwd
> cd <desired-location>

# Clone FASRC User_Codes repository: https://github.com/fasrc/User\_Codes
HTTPS - git clone https://github.com/fasrc/User\_Codes.git

# In <desired-location>, go to GenAI-Tools folder in User_Codes:
> cd User_Codes/Training/GenAI-Tools
```



- [Training Materials – FASRC DOCS](#) - Slide Deck

Sandboxing

- Agent harness permissions are generally enforced with respect to the working directory the agent harness was launched in, with access outside requiring approval
 - *Do not use \$HOME*
- *Sandboxes* use OS-level mechanisms to *enforce* filesystem / network restrictions
- Support:
 - Claude Code for VS Code
 - not supported (<https://github.com/anthropics/claude-code/issues/64061>)
 - Use Claude Code (CLI) if needed (or dev container)
 - Codex (VS Code)
 - enabled by default (<https://learn.chatgpt.com/docs/sandboxing>)
 - Posit Assistant
 - `/sandbox` command not supported in FASRC OOD RStudio Server
 - Jupyter AI
 - Not supported

Jupyter-AI

- <https://jupyter-ai.readthedocs.io/>
- JupyterLab extension that integrates command-line AI coding agents (e.g., Claude Code, Codex, GitHub Copilot)
- AI Coding agents must be separately installed/configured for Jupyter AI to use



Launch JupyterLab (OOD)

1. Open <https://rcood.rc.fas.harvard.edu>
2. Interactive Apps >
Jupyter notebook / Jupyterlab
3. Ensure “Use JupyterLab instead of Jupyter Notebook” is checked
4. Set the working directory (Jupyter root directory) to the directory containing your notebooks/data

Jupyter notebook / Jupyterlab

This app will launch [Jupyter](#) on a compute node on the [FAS-RC](#) cluster:

Partition

test

Partition where to submit your job (--partition).

You can find here the [list of public partitions](#).

☒ Use JupyterLab instead of Jupyter Notebook?

Working Directory (Optional)

/n/netscratch/jharvard_lab/Lab/jharvard

Path to a directory to launch Jupyter server in. Leave blank to use your home directory.

Entering a non-existent directory will cause the job to fail

Jupyter AI - Install AI Coding Agents

- **Claude Code** (<https://code.claude.com/docs/en/quickstart>)

```
$ curl -fsSL https://claude.ai/install.sh | bash
$ claude auth login
...copy/paste URL into browser, log in with HarvardKey...
Paste code here if prompted > ...paste code from browser...
```

- **Codex CLI** (<https://learn.chatgpt.com/docs/codex/cli#getting-started>)

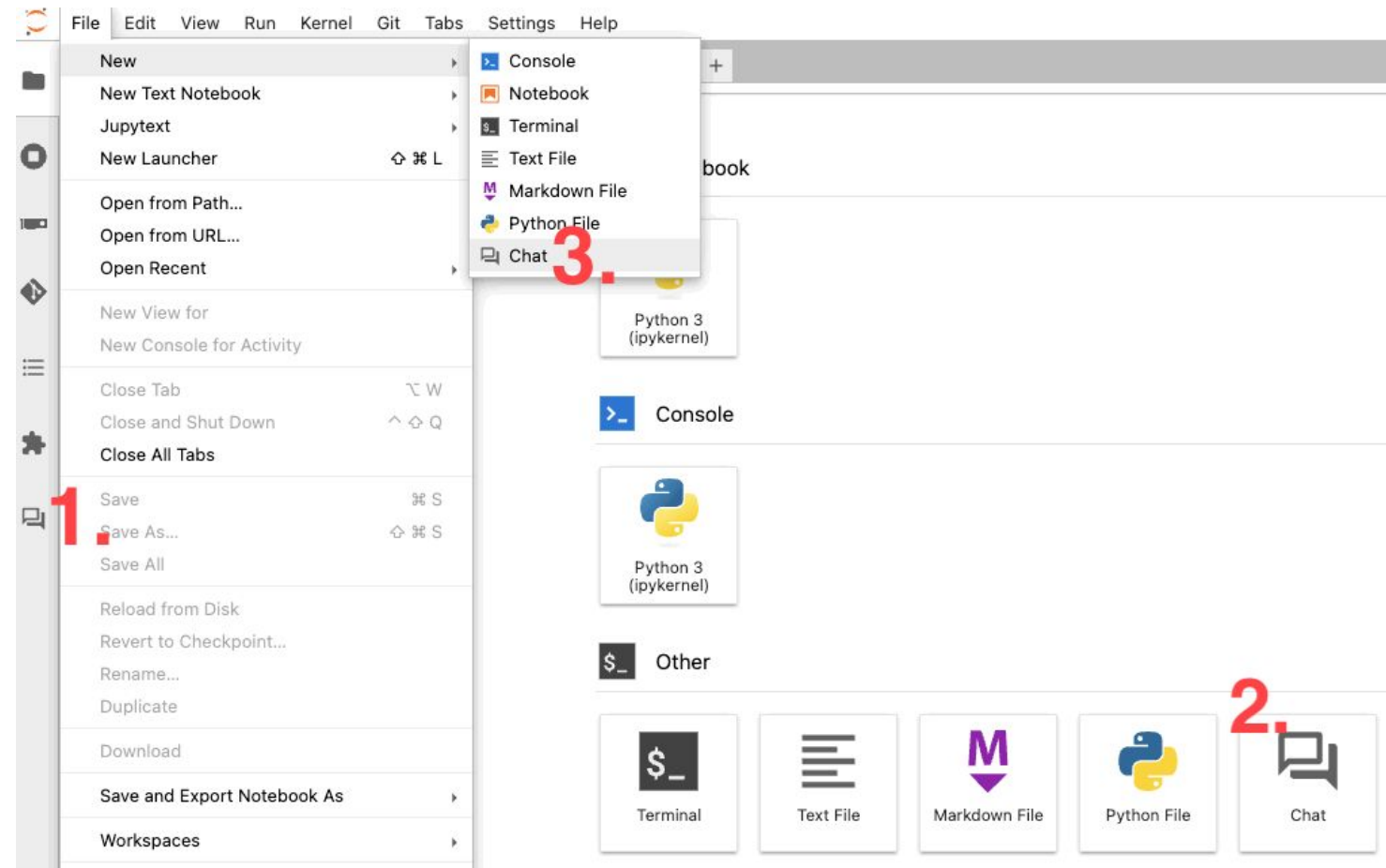
```
$ curl -fsSL https://chatgpt.com/codex/install.sh | sh
$ codex login --device-auth
...follow instructions, logging into ChatGPT EDU account
(harvard-university-chatgpt workspace)...
```

Jupyter AI: Creating Chats

Methods to create a *Chat*:

1. Chat button in the sidebar
2. Chat card in the Launcher
3. File > New > Chat
4. Right click in File Browser pane and select “New Chat” (not pictured)

A `.chat` file will be created in the JupyterLab working directory



Jupyter AI: Codex Permissions Modes

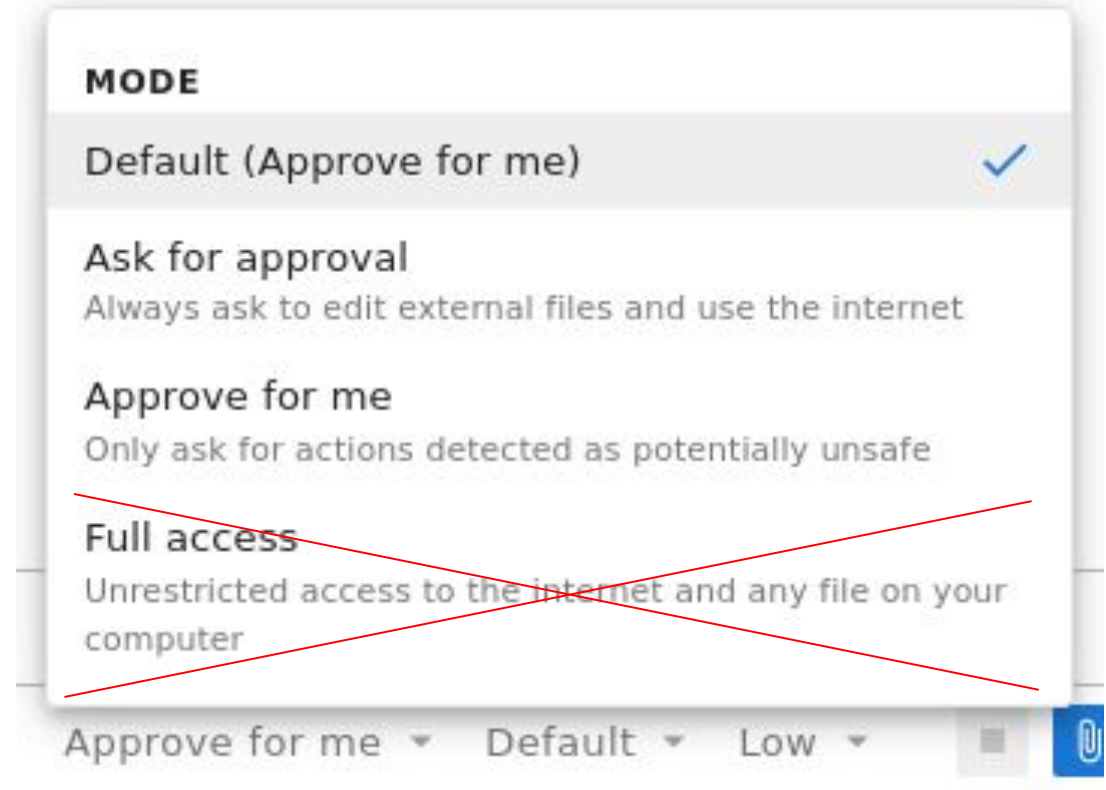
Ask for approval

Codex can read and edit files in the current workspace [default: current working directory] and run routine local commands. It asks before using the internet or going beyond the workspace boundary.

Approve for me

...eligible approval requests are routed through automatic review [by a reviewer agent] instead of showing a prompt for manual approval.

Reference: [Codex] [How Permissions Work](#)



*Do not use **Full access***



Jupyter AI: Claude Permissions Modes

Manual

Reads within the working directory are auto-approved

Accept edits

In the working directory, writes to new/existing files and common filesystem commands (mkdir, touch, rm, rmdir, mv, cp, and sed) are auto-approved

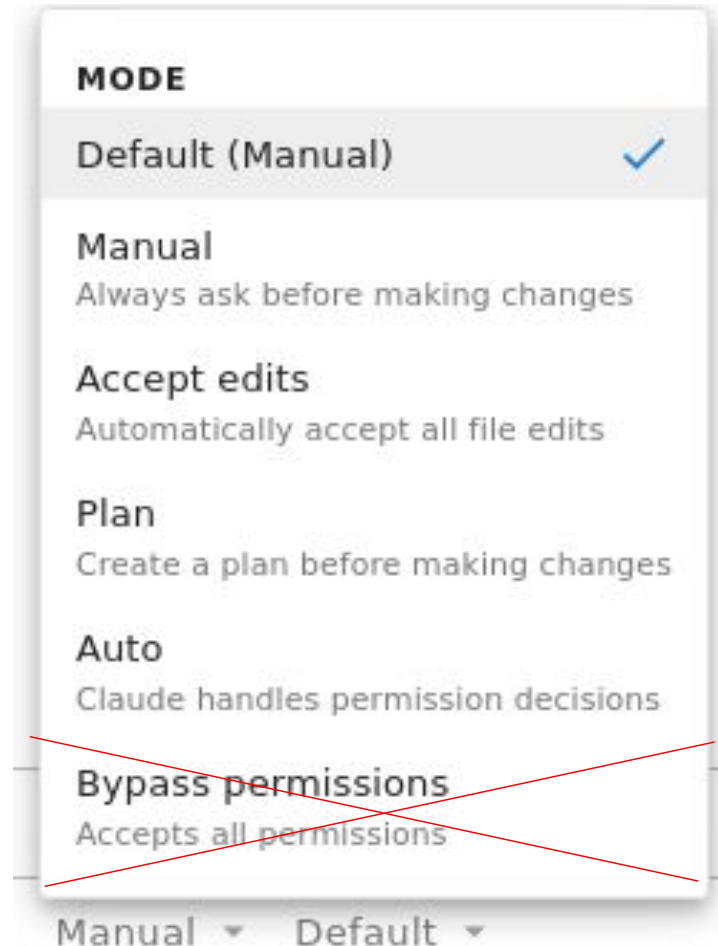
Plan

Reads and several read-only commands within the working directory are auto-approved

Auto

“A separate classifier model reviews actions before they run, blocking anything that escalates beyond your request, targets unrecognized infrastructure, or appears driven by hostile content Claude read.

Reference: <https://code.claude.com/docs/en/permission-modes>



*Do not use **Bypass permissions***

Jupyter-AI Extension using OOD

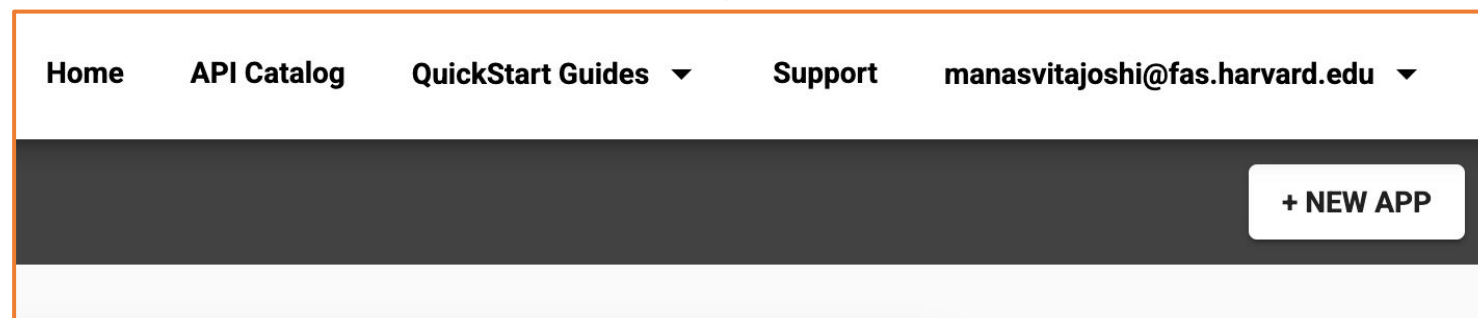
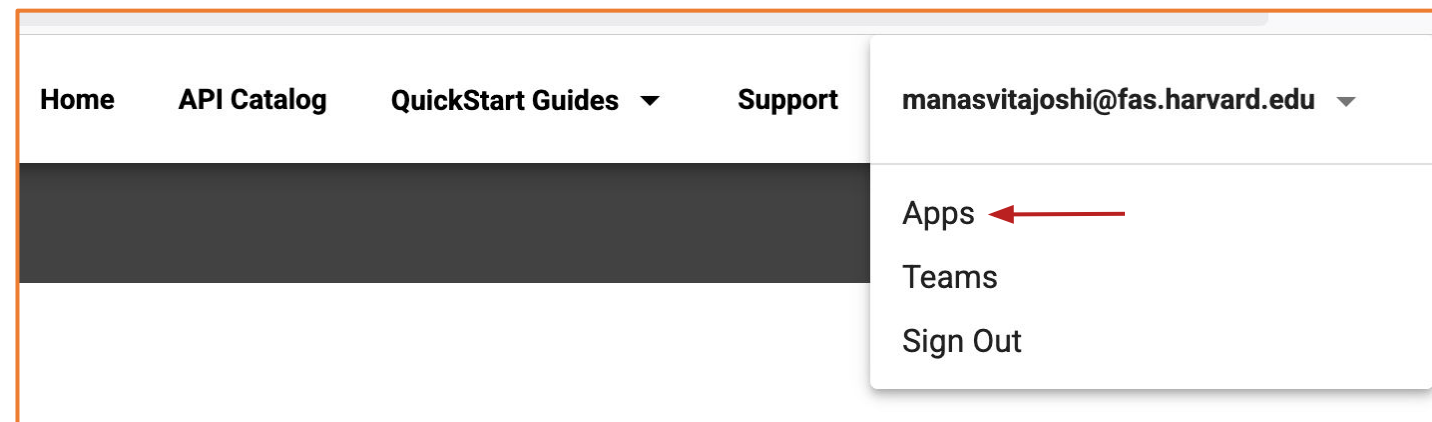
- Accessing OOD from Cannon
 - Connect to FASRC VPN - [Virtual Desktop \(VDI\) through Open OnDemand – FASRC DOCS](#)
 - Then go to <https://rcood.rc.fas.harvard.edu>
- Accessing OOD from FASSE
 - Connect to FASSE VPN - [FASSE VDI Apps – FASRC DOCS](#)
 - Then go to <https://fasseood.rc.fas.harvard.edu>
- Set FASSE proxy: [FASSE Proxy Settings – FASRC DOCS](#)

RStudio Server - GenAI Tools

1. **(Not covered)** GitHub Copilot
 - Original RStudio AI coding assistant
 - [Integrated into RStudio Server IDE](#); supports:
 - Code suggestions (code “auto-complete”)
 - [Questions](#) (simple question/answer prompts)
 - For Harvard use, requires GitHub Copilot for Business license from HUIT
2. Posit Assistant (<https://assistant.posit.co/>)
 - Newer “data analysis assistant”
 - Supports several model [providers](#), including local LLM servers (e.g., Ollama)
 - Claude, ChatGPT EDU subscriptions *not* supported – API key required
 - Posit AI provider is not supported for Harvard use
 - Available in RStudio Server, Posit IDE (*not covered*), and CLI (*not covered*)

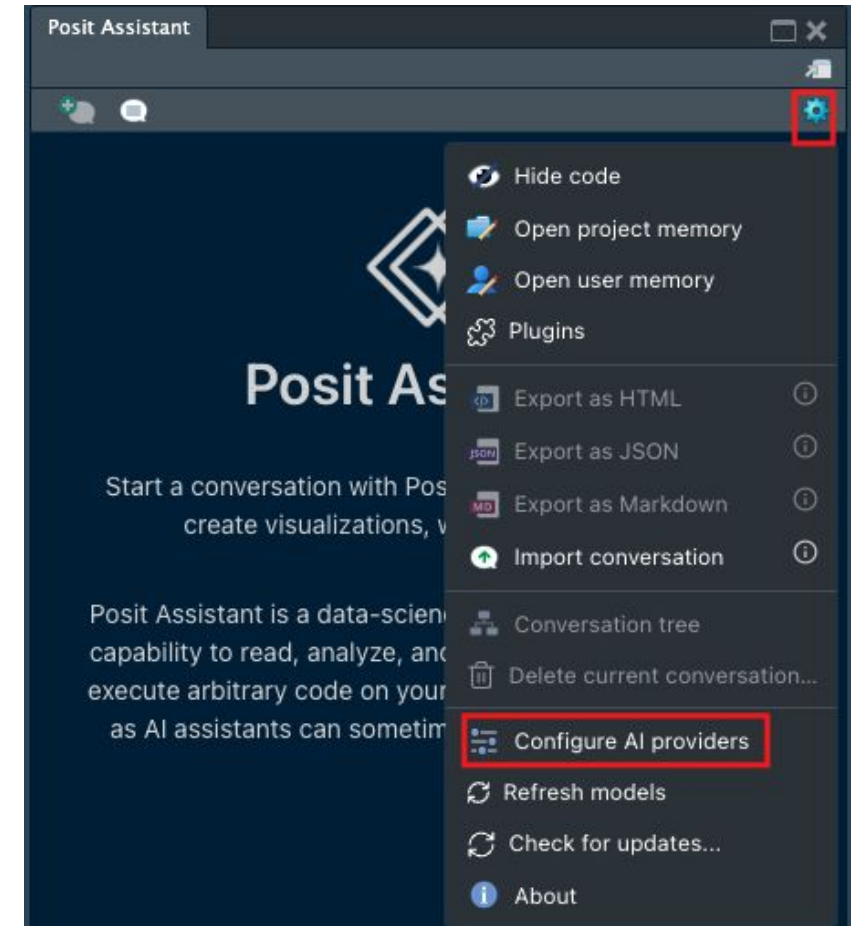
OpenAI for Community Developers API key

- For FAS users - \$10 credit per month
- Harvard Key sign in required:
<https://portal.apis.huit.harvard.edu/docs/ais-openai-direct-limited/1/overview>



RStudio Server: Configure Posit Assistant

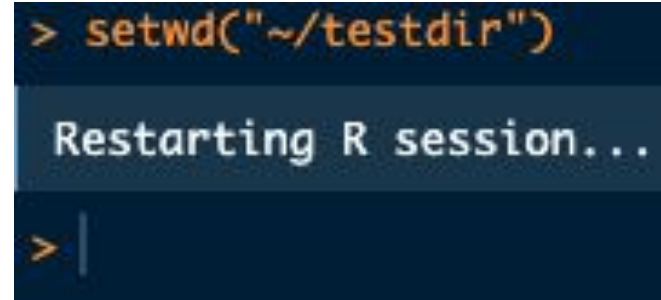
1. Posit Assistant window
 - > Settings (gear icon)
 - > Configure AI Providers
 - > OpenAI Compatible **Connect**
2. **API key:** <your API key from previous slide>
Base URL:
<https://go.apis.huit.harvard.edu/ais-openai-direct-comdev/v2>
3. **Save**
 - a. *Ignore the **Test** button*
4. Close



Posit Assistant - Set Workspace Directory

- When Posit Assistant is started, its *workspace directory* is the R session's working directory
- To change:
Session > Set Working Directory
OR (in R console)
> setwd("/some/directory")

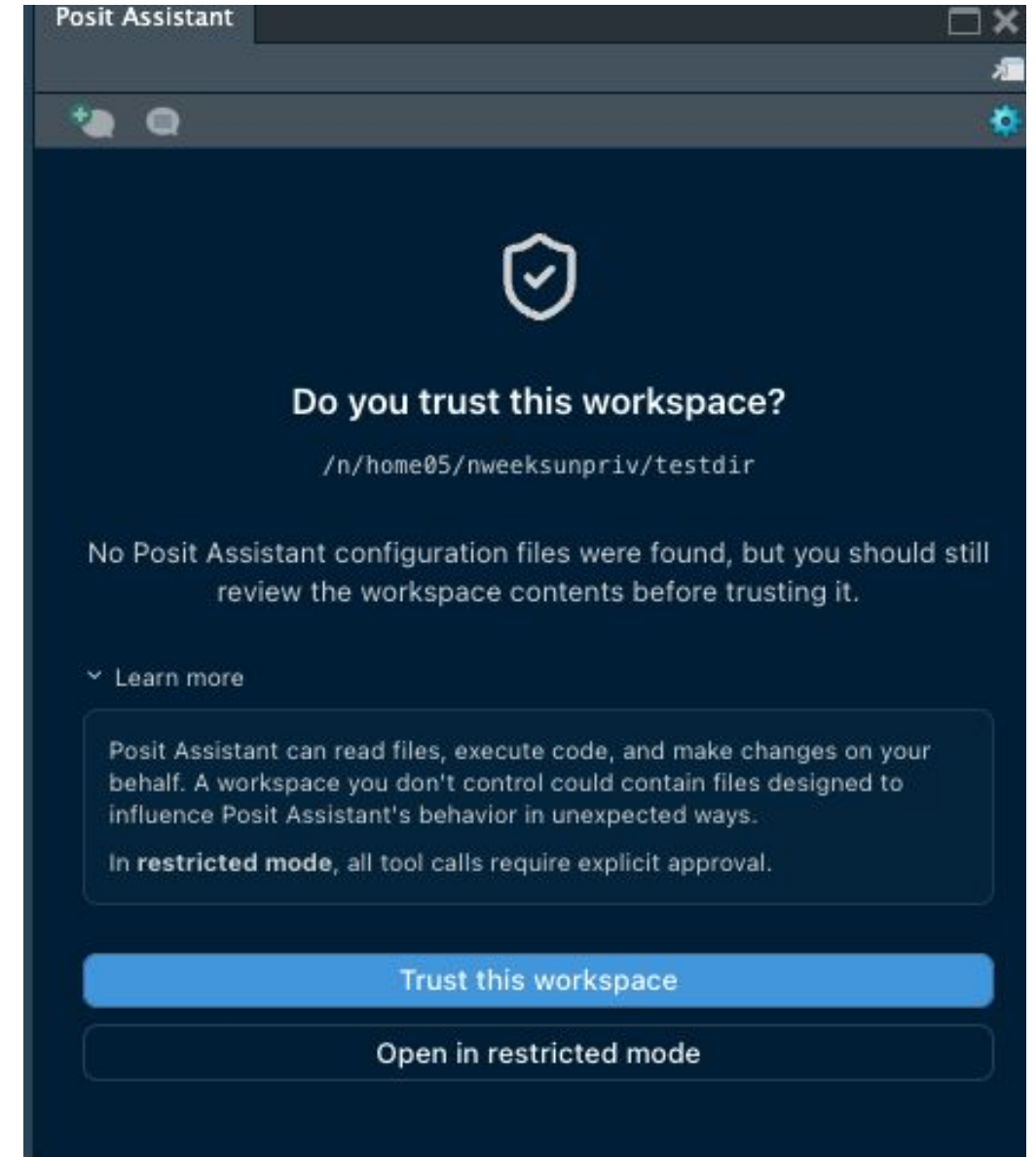
AND Session > Restart R



```
> setwd("~/testdir")  
Restarting R session...  
> |
```

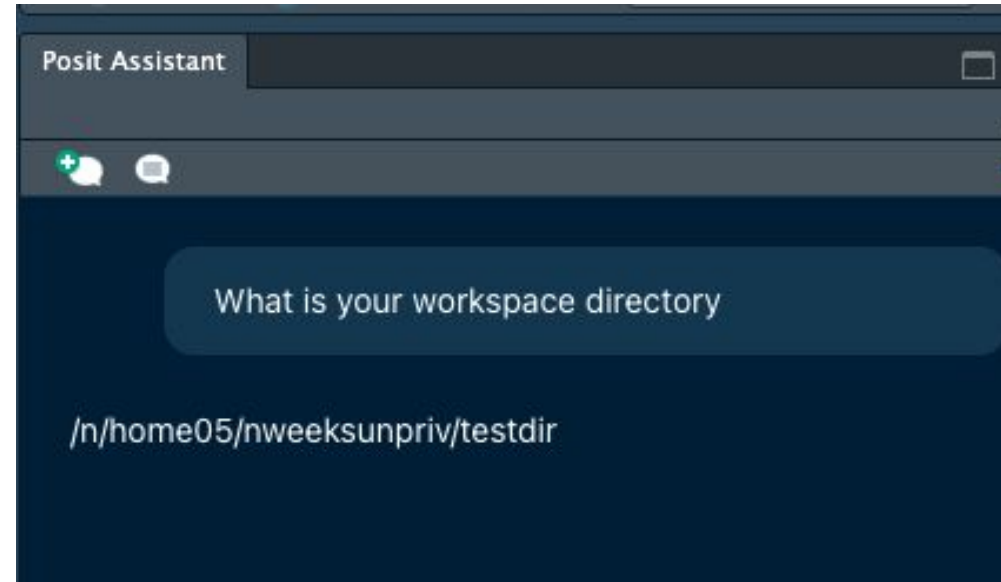
Posit Assistant - Set Workspace Directory

- Will be prompted to *Restart Posit Assistant*.
- If it's the first time Posit Assistant is used with a particular workspace directory, will be prompted to trust the workspace or open in *restricted mode*
 - If the workspace directory has been used before, the last chat session in that workspace should appear



Posit Assistant - Set Workspace Directory

- To verify: ask Posit Assistant what its workspace directory is



RStudio Server - Permissions Modes

Normal

Ask for permission to run “risky” commands

Plan mode

<https://assistant.posit.co/docs/features/plan-mode/>

Auto mode

“A classifier model reviews risky tool calls and allows or blocks them”

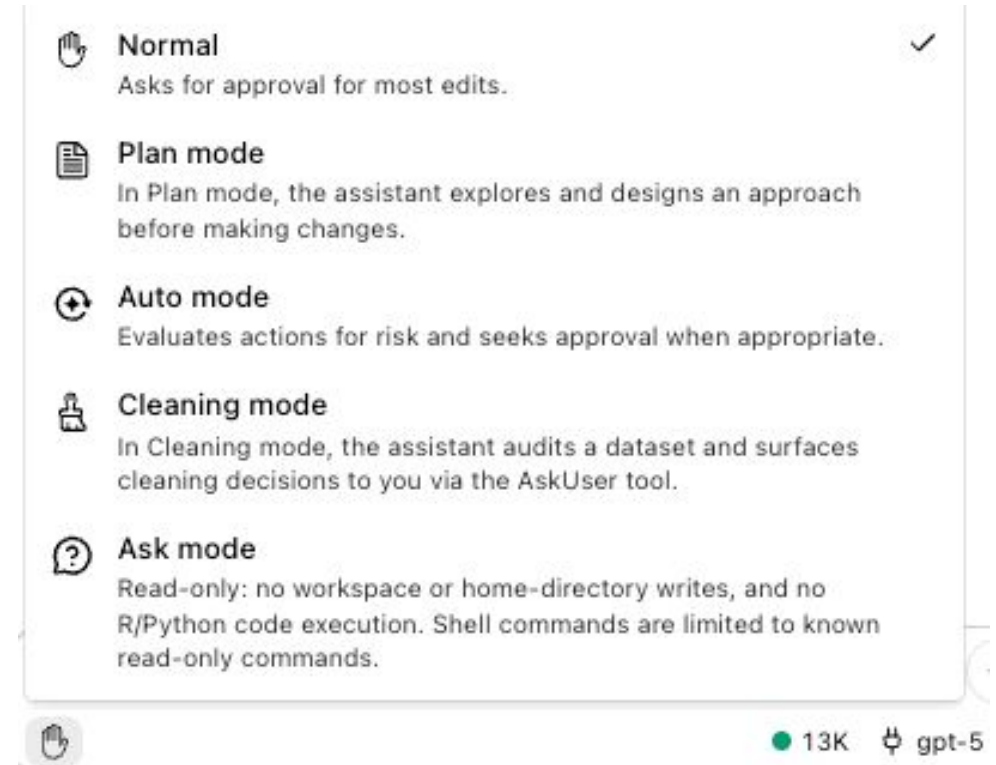
<https://assistant.posit.co/docs/features/permissions/#auto-mode>

Cleaning mode

https://opensource.posit.co/blog/2026-05-08_ai-newsletter/

Ask mode

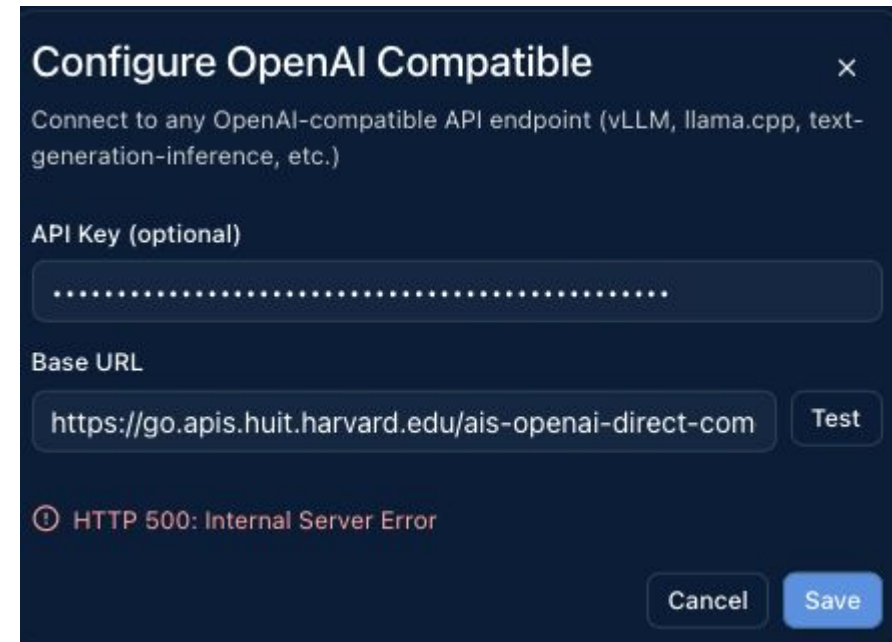
<https://assistant.posit.co/docs/features/ask-mode/>



No YOLO mode 👍

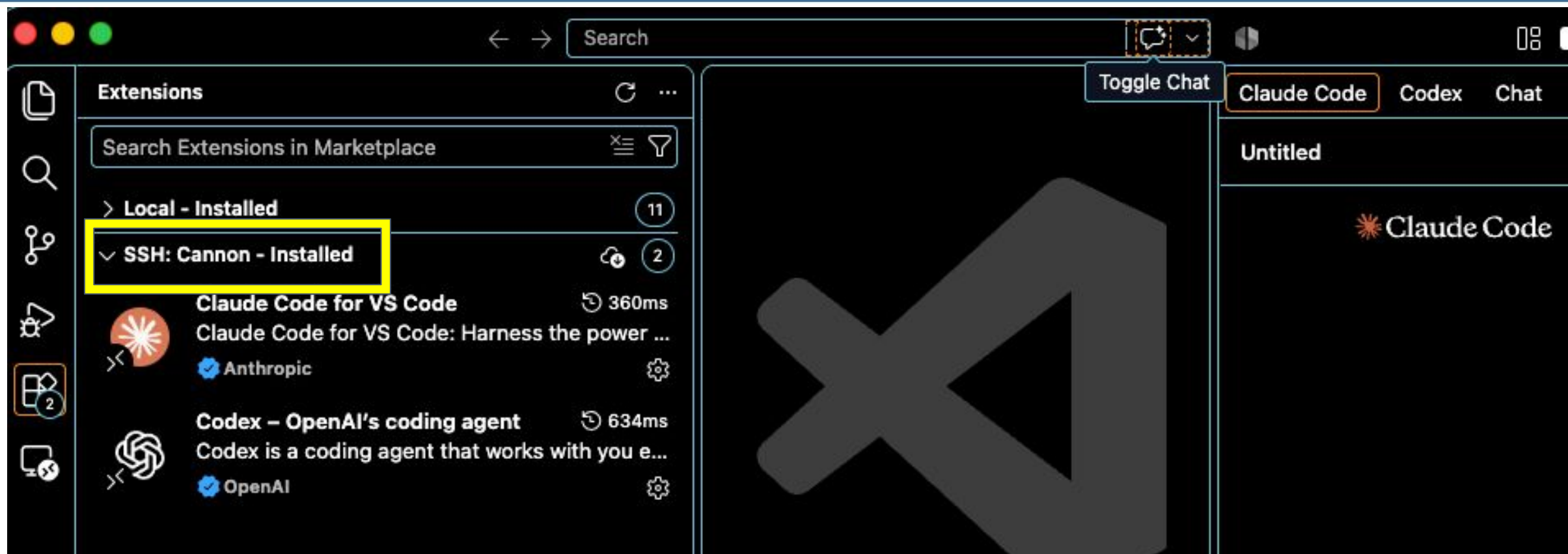
RStudio Server - Posit Assistant issues

- Posit Assistant /sandbox does not work in FASRC OOD Rstudio Server
- gpt-5.6-* (and gpt-6-astra) models currently do not work (as of Posit Assistant 1.3.1)
<https://github.com/posit-dev/assistant-feedback/issues/92>
- “Test” button results in an HTTP 500 error when using the HUIT *OpenAI for Community Developers* endpoint (pictured)



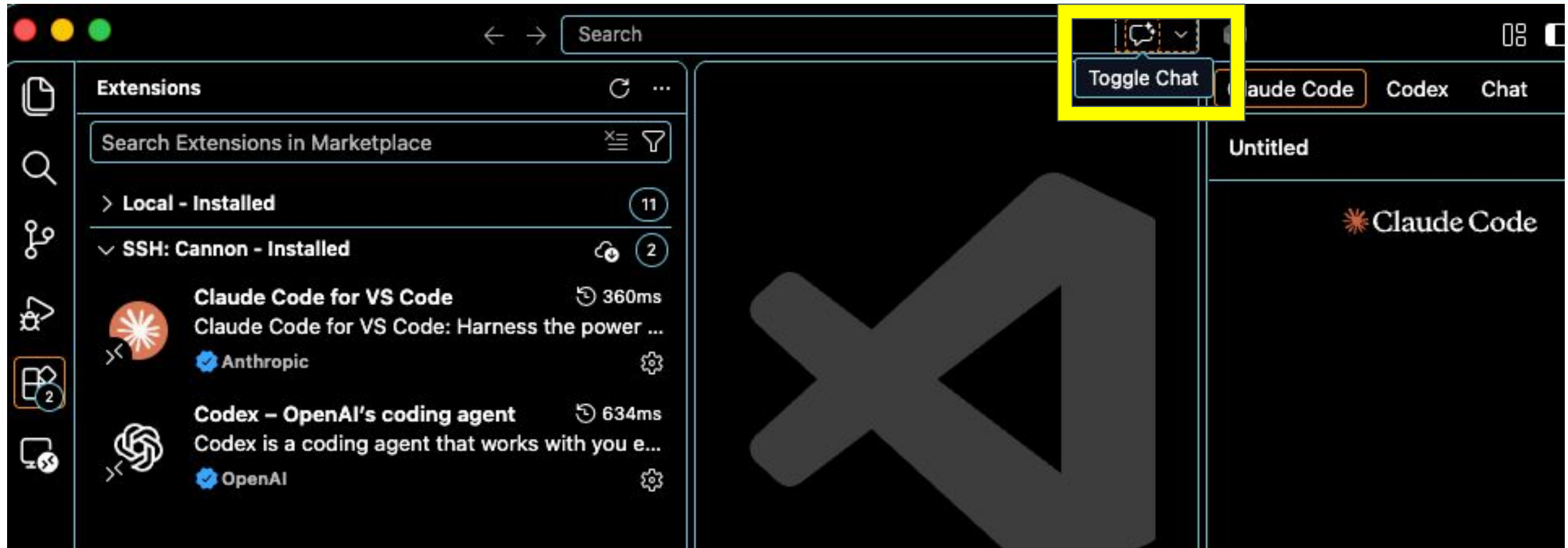
VS Code

- [Codex](#), [Claude Code for VS Code](#) extensions
 - Bundle Codex & Claude Code CLIs, no separate installation needed/used
- [Chat](#) (not covered)
 - Uses GitHub Copilot by default

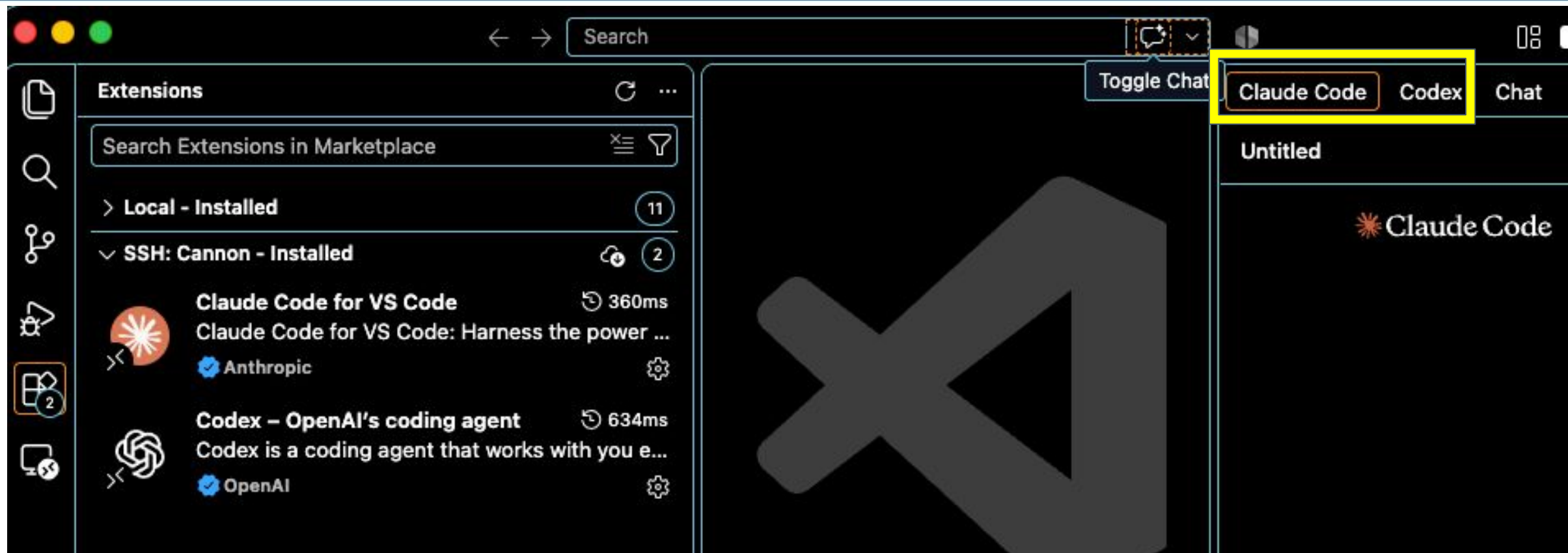


1. Install *Claude Code for VS Code* (by Anthropic) and/or *Codex - OpenAI's coding agent* (by OpenAI) extensions are installed *in the remote* (SSH or Tunnel)

Avoid similarly-named extensions by other authors



2. Click “Toggle Chat” button to open Chat window (if not already open)



3. Select “Claude Code” or “Codex” tabs in the Chat window to sign in / use

Chat tab is (primarily) for GitHub Copilot – don’t use unless your GitHub account is associated with a GitHub Copilot Business subscription from HUIT

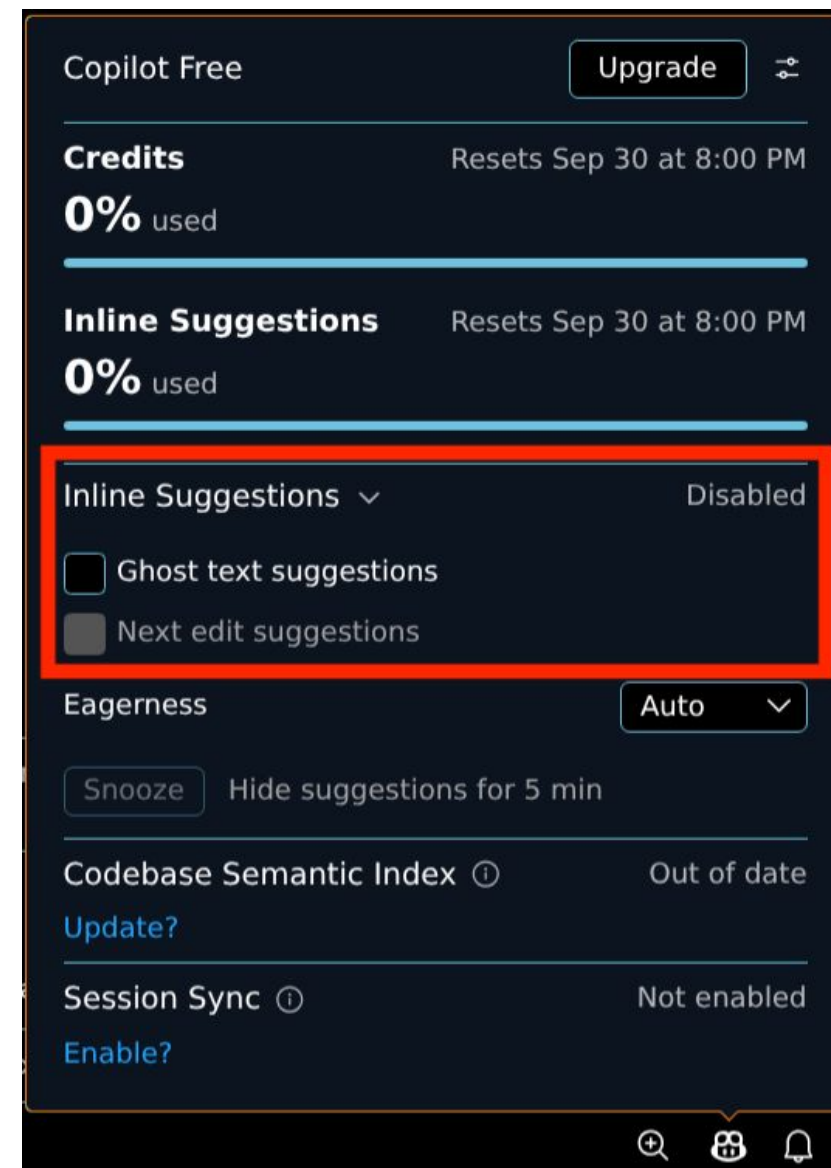
VS Code - Inline Suggestions

Disable Inline Suggestions if you do not have a GitHub Copilot Business license

If signed into a GitHub account (that has GitHub Copilot enabled), Inline Suggestions (enabled by default) causes:

The surrounding code from your cursor position is pre-processed by the inline suggestions system, combined with contextual information (such as code snippets from open tabs in the editor), and sent to a large language model in the form of a prompt.¹

To disable: click GH Copilot icon in bottom right corner of VS Code window and uncheck “Ghost text suggestions”

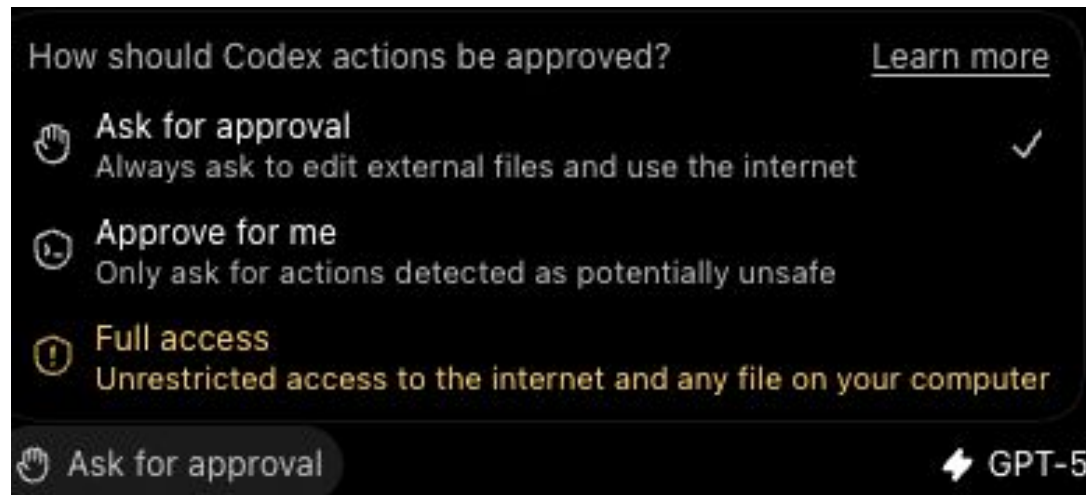


1. <https://docs.github.com/en/copilot/responsible-use/inline-suggestions#6-performance>

VS Code - Agent Permission Modes

Codex

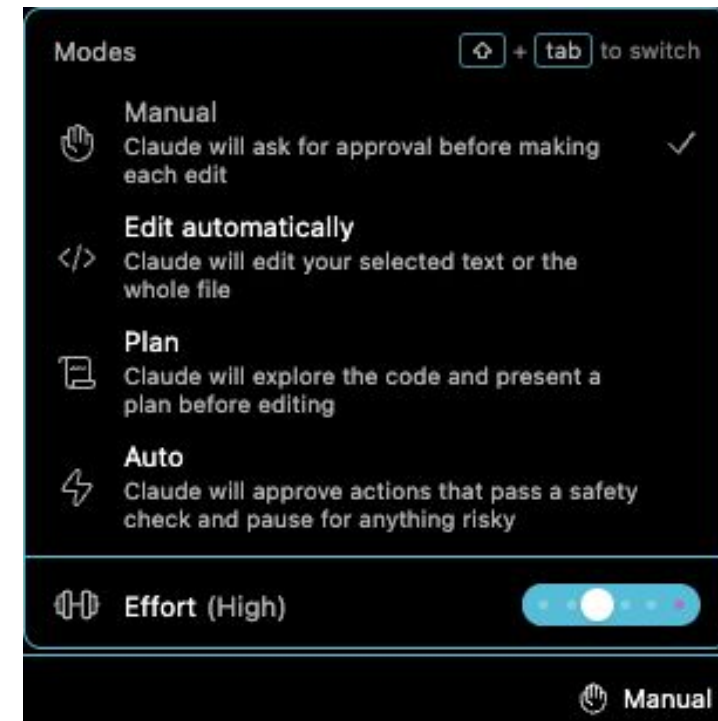
Same as with Jupyter AI
Avoid full access mode!



*Permissions are with respect to VS Code workspace (> **Open Folder** to change)*

Claude Code for VS Code

Same as with Jupyter AI – except
no *Bypass permissions* mode 👍



Dev Containers

- [VS Code Dev Containers extension](#) “lets you use a Docker container as a full-featured development environment”
- A way to sandbox AI coding assistants / agents
 - Only repository with devcontainer.json is accessible by default

See [Using Dev Containers on the FASRC Cluster](#)

Example:

User_Codes/Training/GenAI-Tools/devcontainer.json

After configuring VS Code settings:

```
sbatch devcontainer.job
```

GitHub Copilot Business

- Request by sending ticket to HUIT
 - Once added to harvard-copilot-users GitHub org, [GitHub Copilot settings](#) managed by HUIT
- Must use GitHub Copilot Business from HUIT on FASRC clusters
 - GitHub Copilot Individuals (Free/Pro/Pro+/Max) is for level 1 data only
- Has models not in Claude or Codex
 - Gemini, Grok, MAI
 - Often has latest GPT models before they are available in ChatGPT EDU
- Integrated with GitHub Web UI
 - Ask questions of a repository, troubleshoot GitHub Actions, assign to pull requests
- *No individual quota*: pooled monthly credits shared among all harvard-copilot-users members
 - Models are more “expensive” than Claude / ChatGPT EDU subscriptions

Google Antigravity / Gemini Code Assist

- Harvard's Google Workspace for Education (g.harvard.edu) does *not* include enterprise data protection for Google Antigravity (i.e., Level 1 data only)
- Contact HUIT for enterprise licensing options
- See HUIT article [Enterprise-Grade Data Protection for Google Gemini](#)

FASRC documentation

- FASRC docs: [FASRC DOCS](#)
- Training
 - Calendar: [Training Calendar | FAS Research Computing](#)
 - Material: [Training Materials – FASRC DOCS](#)
- Getting help
 - Office hours: [Virtual Office Hours | FAS Research Computing](#)
 - Ticket:
 - HUIT Service Portal -> Submit Ticket: [Submit Ticket - IT Help](#)
 - Email: rchelp@rc.fas.harvard.edu

Upcoming Trainings

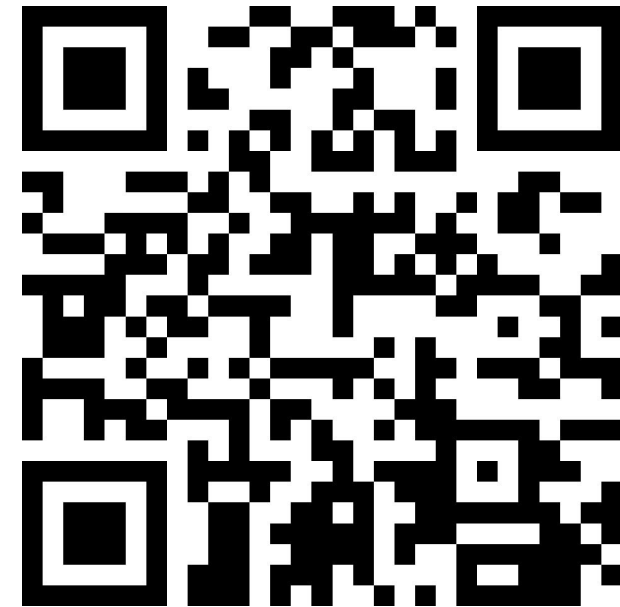
Training calendar: [Training Calendar | FAS Research Computing](#)

DATE	TIME	SESSION / TOPIC
Thursday, Sept 17	1:00 pm - 2:00 pm	FASRC: Getting started on the FASRC clusters (intro)
Wednesday, Sept 23	12:00 pm - 1:00 pm	Research Data Management: Concepts, Tools, and Strategies
Thursday, Sept 24	1:00 pm - 2:00 pm	FASRC: Getting started on FASRC clusters with Open OnDemand
Friday, Sept 25	1:00 pm - 2:00 pm	FASRC: Getting started on FASRC clusters with CLI

Training session evaluation

Please, fill out our training session evaluation. Your feedback is essential for us to improve our trainings!!

<https://tinyurl.com/FASRC-training>





Thank You!
Questions? Comments?

Supplemental Content

Ollama - Open Source LLMs

- AI tool to download, run, & manage open source LLMs locally
- Why run local?
 - **No API costs, total data privacy**
- On the cluster, can be run via Singularity on a compute node
 - Build Singularity container image from Docker Hub container image
 - Spin up the Ollama server in one terminal
 - Open another terminal and ssh to the same host
 - Run ollama on open source LLM of your choice
- https://github.com/fasrc/User_Codes/blob/master/AI/AITools/ollama.md
- Another tool: [vLLM](#)

Permissions modes - VS Code Chat

- *Avoid “Bypass Permissions”*
 - Possible exception: using a dev container to isolate filesystem access

