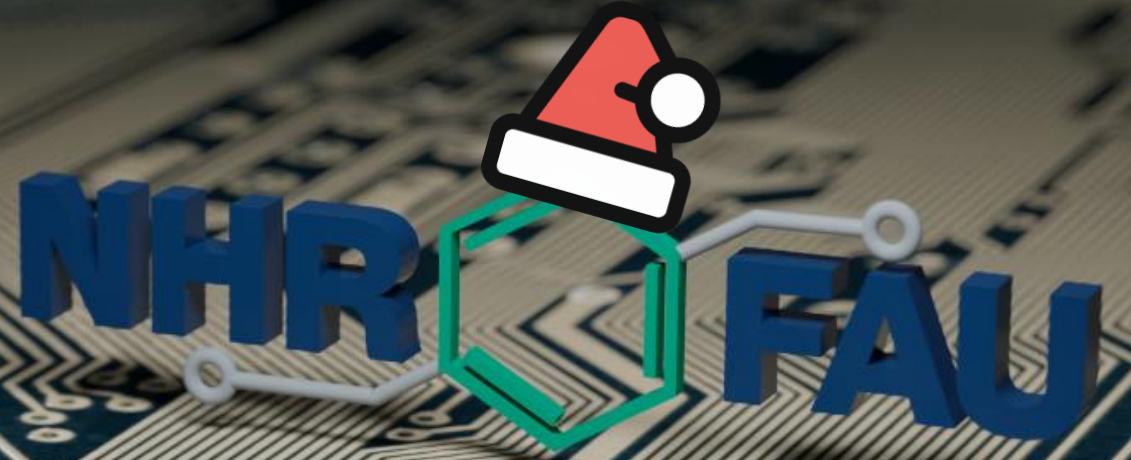


NHR@FAU HPC Café
December 16, 2025



Using Visual Studio Code Efficiently

Chang Liu

Erlangen National High Performance Computing Center (NHR@FAU)

Visual Studio Code (VS Code)



Diverse extensions:

- Python, Jupyter, Container, etc.
- Most are useful 😊

Light-weighted IDE:

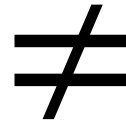
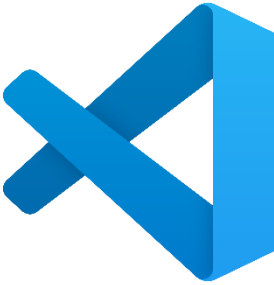
- From Microsoft
- Code editor and terminal
- Built-in debugger

Powerful IntelliSense:

- Various language servers.
- Copilot

Extra support:

- Git
- Remote development



Visual Studio

Today's topic:

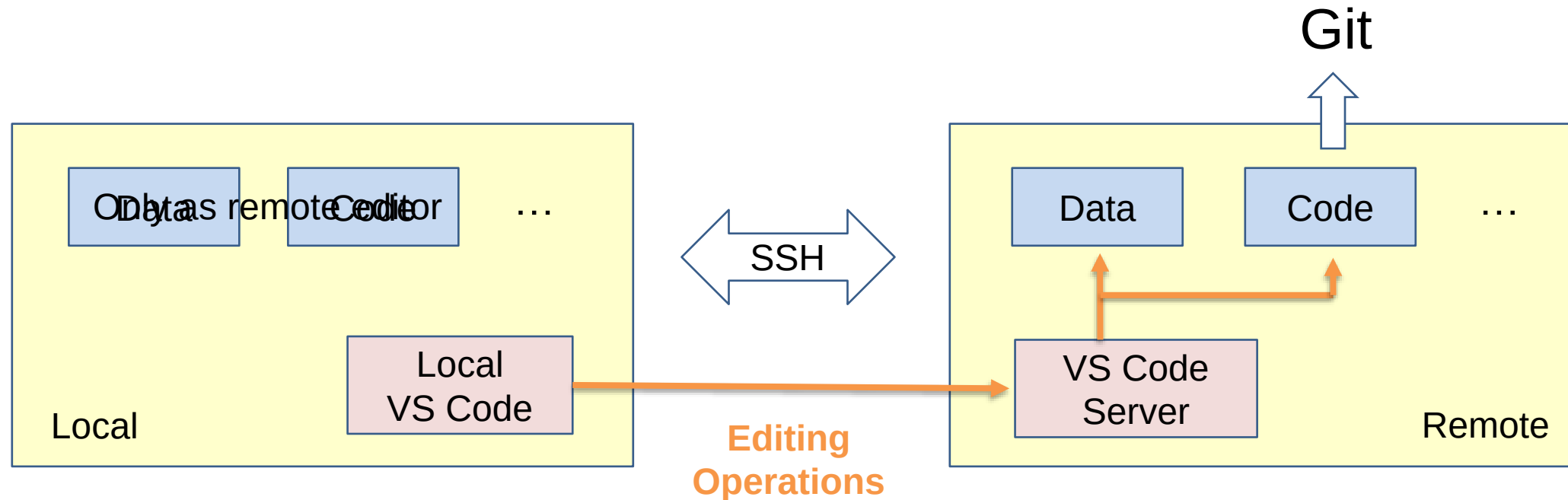
- Remote development
- Copilot

Using VS Code on HPC systems: Remote Development



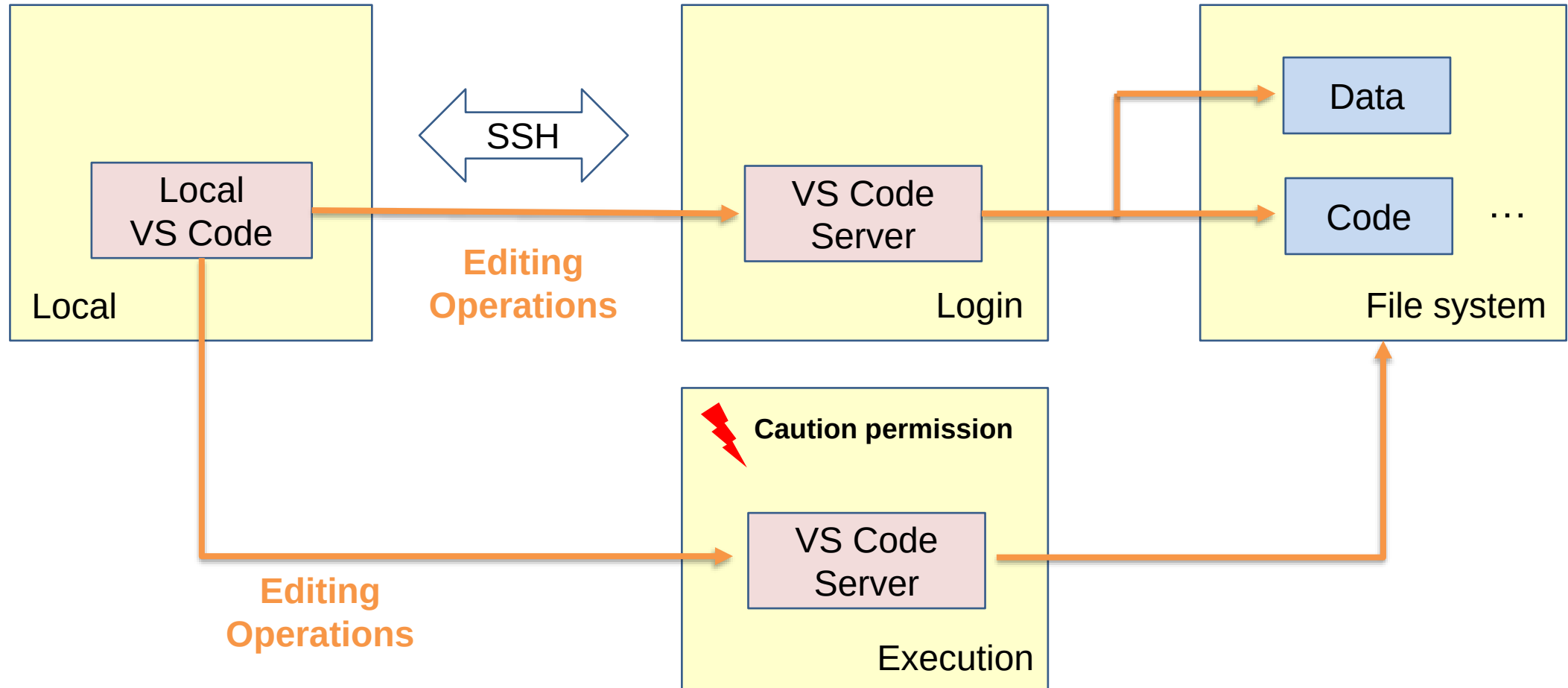
Connect to HPC via VS Code

- **VS Code cannot configure the SSH automatically.** Follow our doc!
- How remote development works with VS Code...



Connect to HPC via VS Code

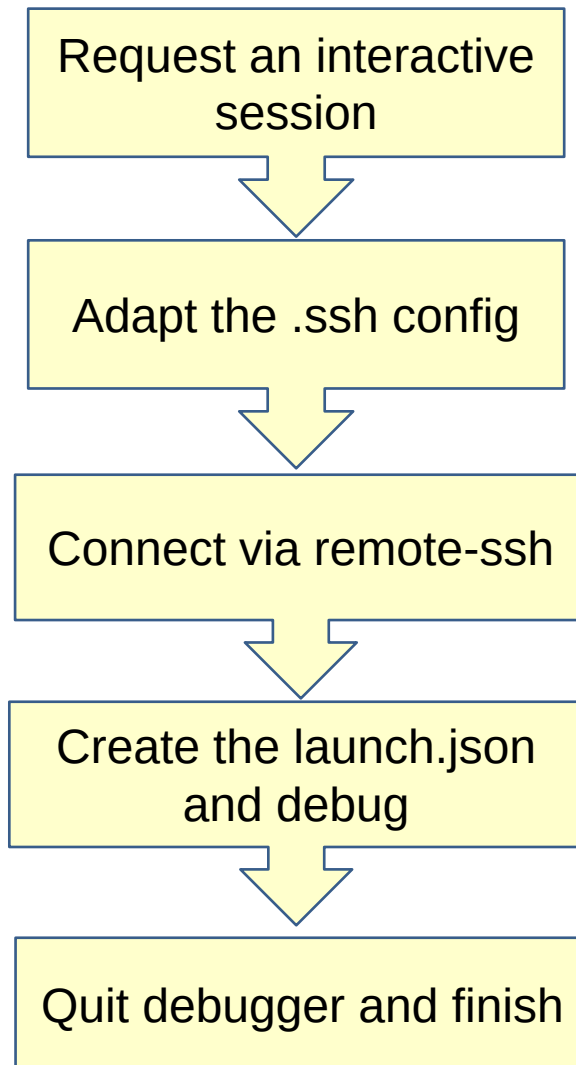
- HPC systems are more complicated...



Connect to HPC via VS Code

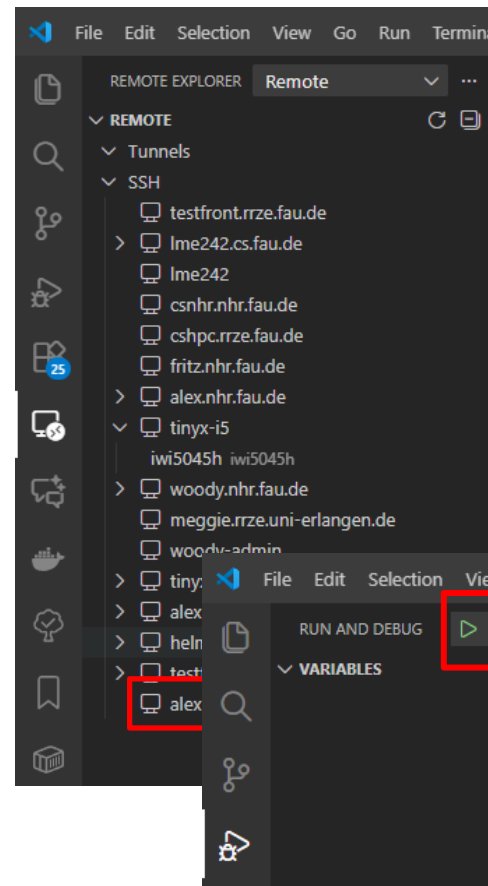
- Connection to the **login node**:
 - Access to the file system.
 - Edit on data.
 - Configuration of the environment.
 - Constant hostname.
- Connection to the **execution node**:
 - Direct access to resources on execution node, i.e. GPUs.
 - Additional SSH configuration needed.

Example: Interactive Python debug in HPC



```
unrz110h@alex2:/home/woody/unrz/unrz110h/nutshell$ salloc --gres=gpu:a100:1 --time=00:30:00
...
unrz110h@a0903:/home/woody/unrz/unrz110h/nutshell$ hostname
a0903.nhr.fau.de
```

```
# needed only for debugging Tier3-Grundversorgung
Host alex-debug
  HostName a0903.nhr.fau.de
  User ...
  ...
```



```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python Debugger: Current File",
      "type": "debugpy",
      "request": "launch",
      "program": "${file}",
      "console": "integratedTerminal",
      "python": "/home/woody/.../bin/python",
    }
  ]
}
```

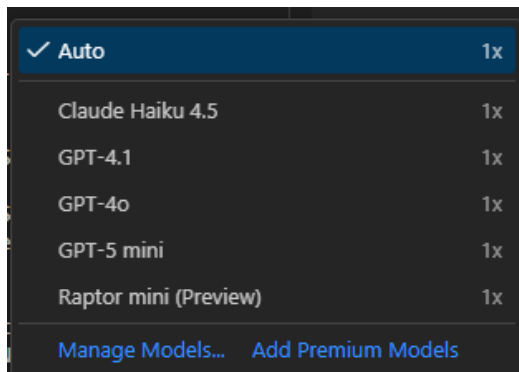
Using VS Code on HPC systems: Copilot



- Copilot is a LLM by GitHub (more than just IntelliSense).
- Training data:

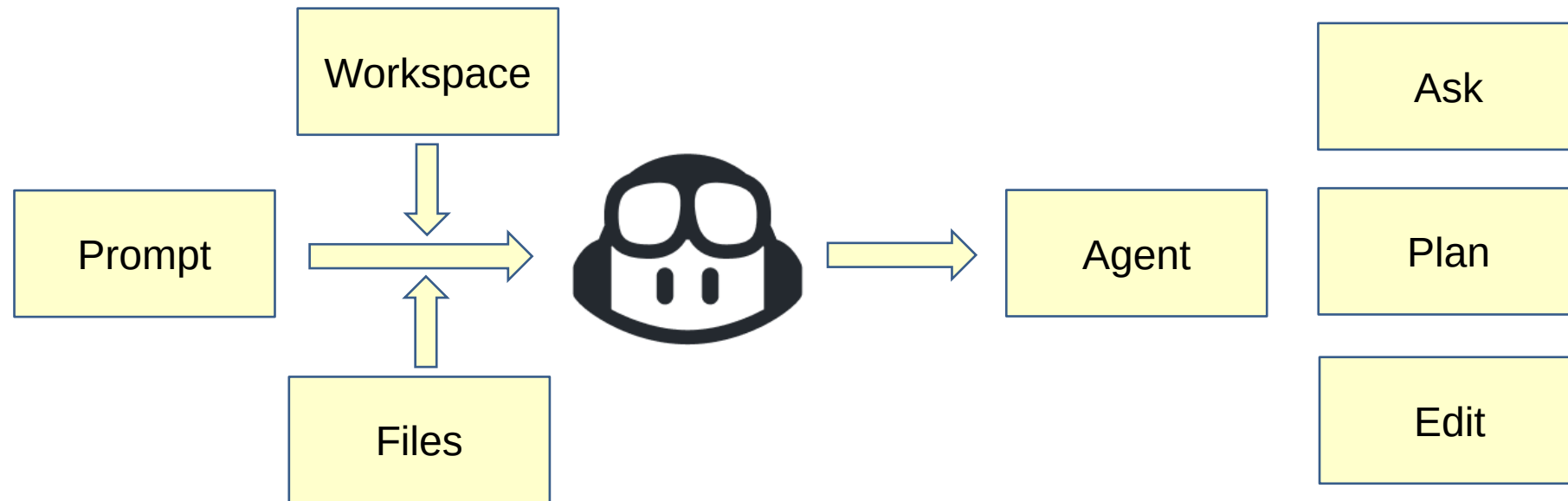
Type	Note
Public GitHub code	Major training data
Public documentation	As natural language text
Forums/Q&A	Likely Stack Overview

- Now 4 backend models, from my account.



Copilot

- Copilot is a LLM by GitHub (more than just IntelliSense).
- Training data...
- How Copilot in VS Code works:



Example: Code generation

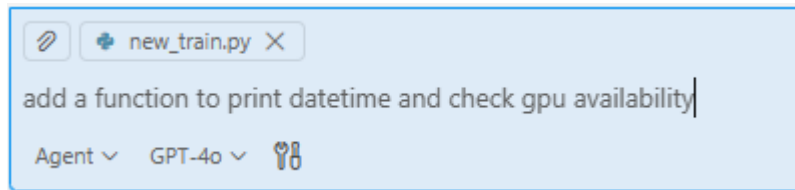
- Copilot generates code in two modes
 - **Inline completion**
 - Via chat

```
18 # function to print datetime and check gpu availability
19 from datetime import datetime
✦ 20 def print_datetime_and_gpu():
    print(f"Current date and time: {datetime.now()}")
    if torch.cuda.is_available():
        print(f"GPU is available: {torch.cuda.get_device_name(0)}")
    else:
        print("GPU is not available.")
```

Generated code

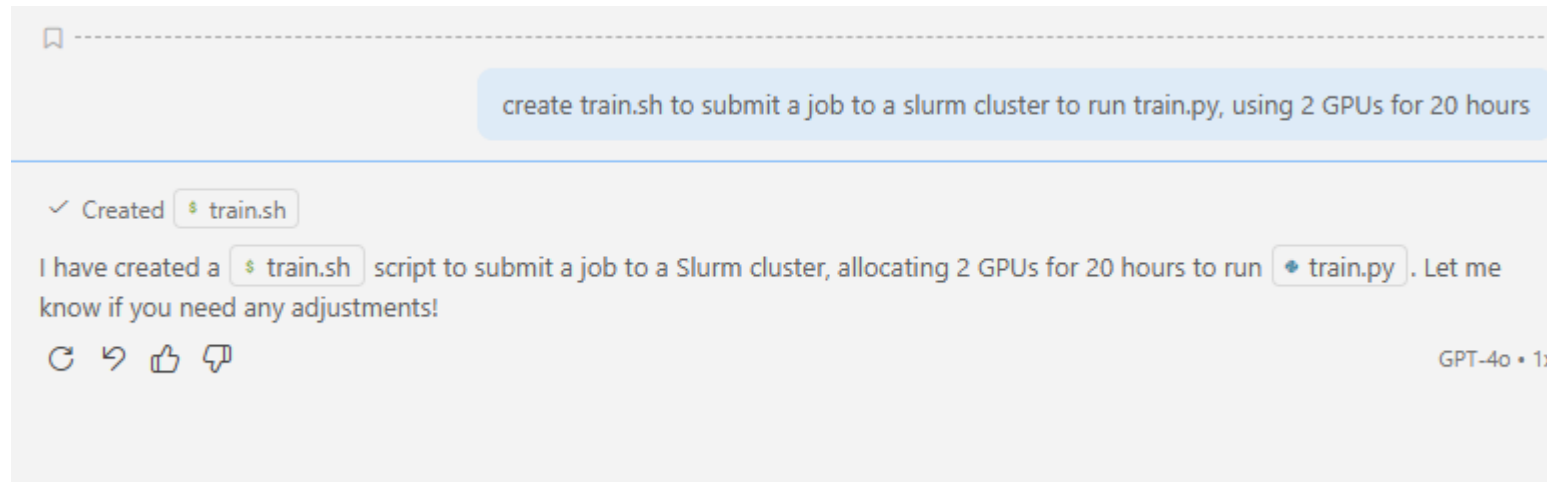
Example: Code generation

- Copilot generates code in two modes
 - Inline completion
 - **Via chat**



```
17 from datetime import datetime
18
19 # function to print datetime and check gpu availability
20 def print_datetime_and_gpu():
21     """Prints the current datetime and checks GPU availability."""
22     print(f"Current datetime: {datetime.now()}")
23     if torch.cuda.is_available():
24         print(f"GPU is available: {torch.cuda.get_device_name(0)}")
25     else:
26         print("GPU is not available.")
27
28 # Example usage
29 print_datetime_and_gpu()
--
```

Example: Copilot generates the job bash



```
#!/bin/bash
```

```
#SBATCH --job-name=train_job
#SBATCH --output=train_job_%j.out
#SBATCH --error=train_job_%j.err
#SBATCH --time=20:00:00
#SBATCH --gres=gpu:2
#SBATCH --cpus-per-task=4
#SBATCH --mem=16G
```

```
# Load necessary modules
module load python/3.8
module load cuda/11.3
```

```
# Activate virtual environment
source ~/venv/bin/activate
```

```
# Run the training script
python train.py
```



Proper Usage of Copilot



- Careful review and validate

Do:

- Generate code with clear prompt.
- Give Copilot a working example.
- Ask Copilot to improve given code.
- Refer to the system docs when errors occur.



- Rely completely on Copilot.

Don't:

- Blindly copy and paste code.
- Assume that the code is optimized or secured.
- Generate the job bash for slurm.

Conclusion

- VS Code is a powerful and flexible IDE
 - With the proper extensions, it becomes much more than a IDE.
- VS Code is an effective UI for HPC systems
 - It enables remote editing, debugging, and environment management.
- More applications and extensions:
 - Container, Jupyter, etc.
- Be aware what AI generates!
 - Copilot significantly speeds up programming, **only with careful review and validation.**