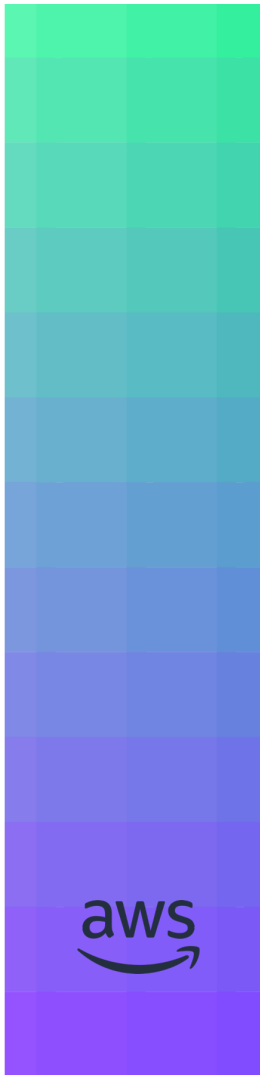
Model Driven Agents:

How AWS Moved Beyond Orchestration with Strands SDK

Nicholas Clegg
Senior Software Engineer - AWS
Strands Agents SDK



Kiro IDE

Q Developer

AWS Bedrock

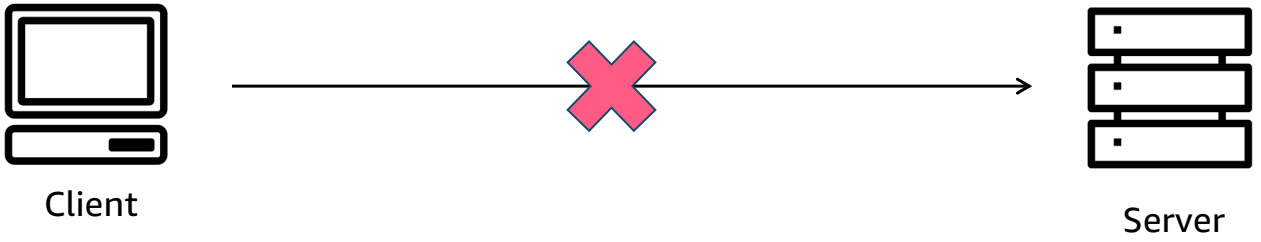
Agenda:

- When Orchestration Breaks Down
- The Model-Driven Approach
- Your First Strands Agent
- Multi and Meta Agents
- Takeaways

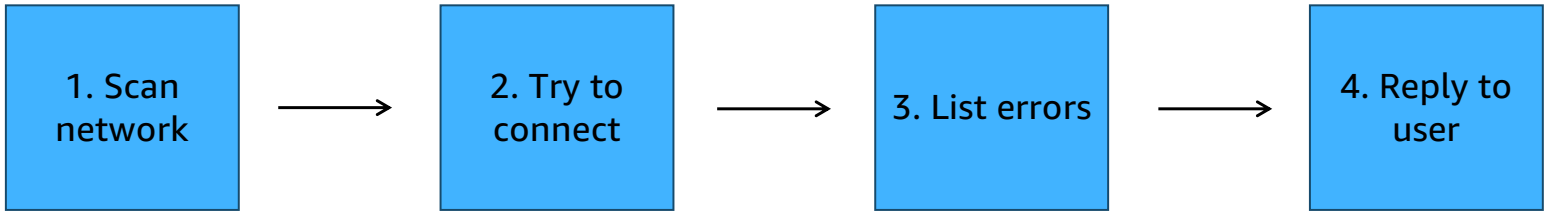
When Orchestration Breaks Down

The Problem: Analyzing a Network

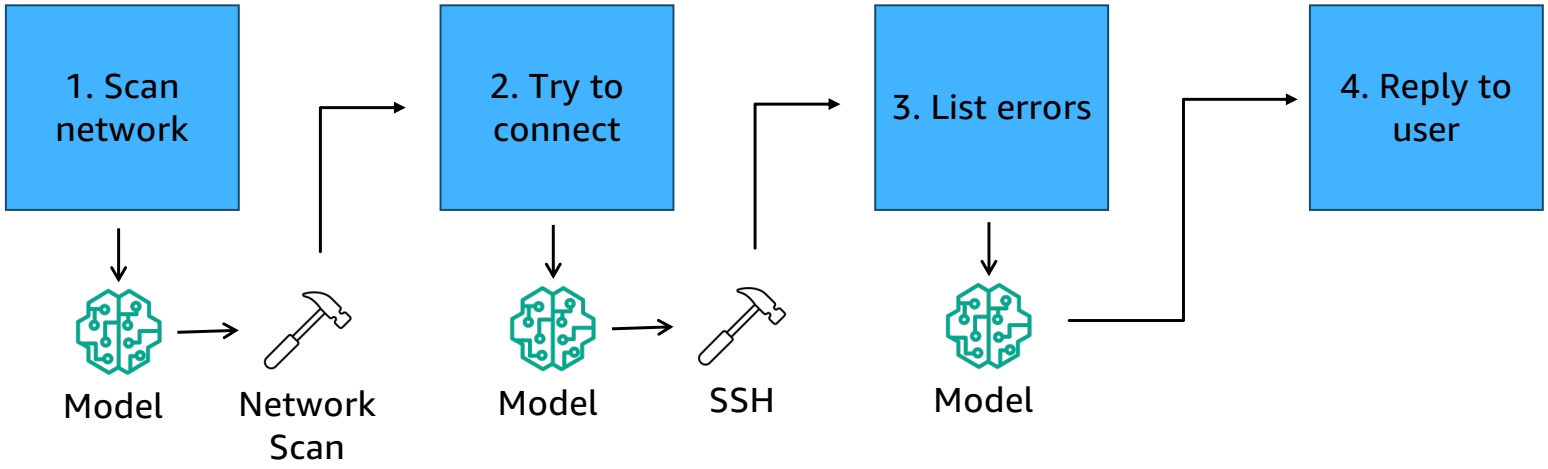
- Customers had a hard time debugging network connections
- Build an Agent that can analyze network issues

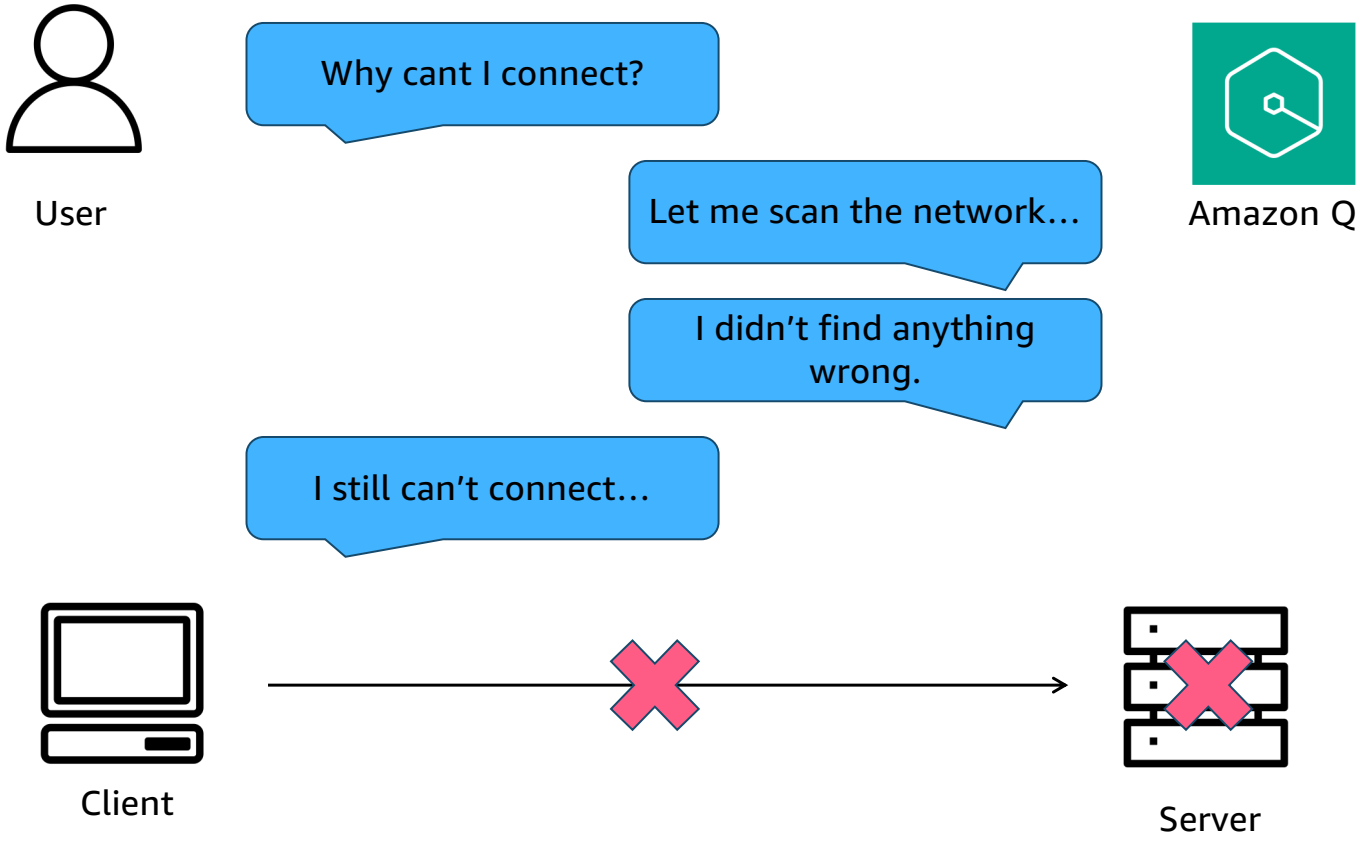


Network Analyzer Design - Workflow



Network Analyzer Design - Workflow

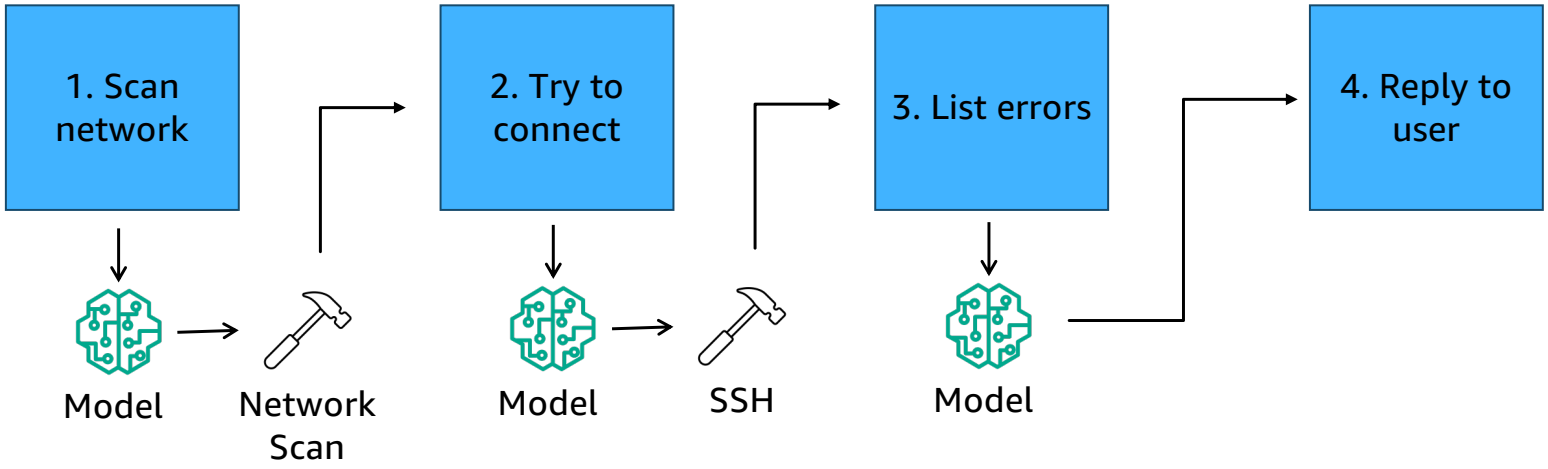




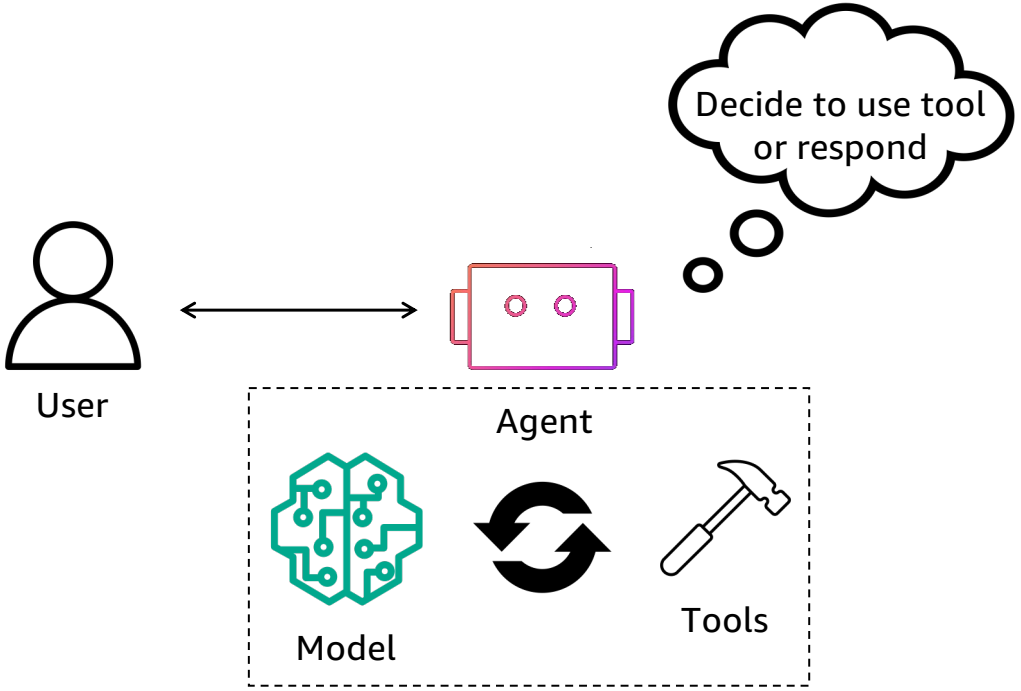
Network Analyzer Issues

- Small number of “success” cases
- Complex state machines and error handling
- Frustrating error messages

Network Analyzer Design - Workflow



Network Analyzer Design – Model Driven

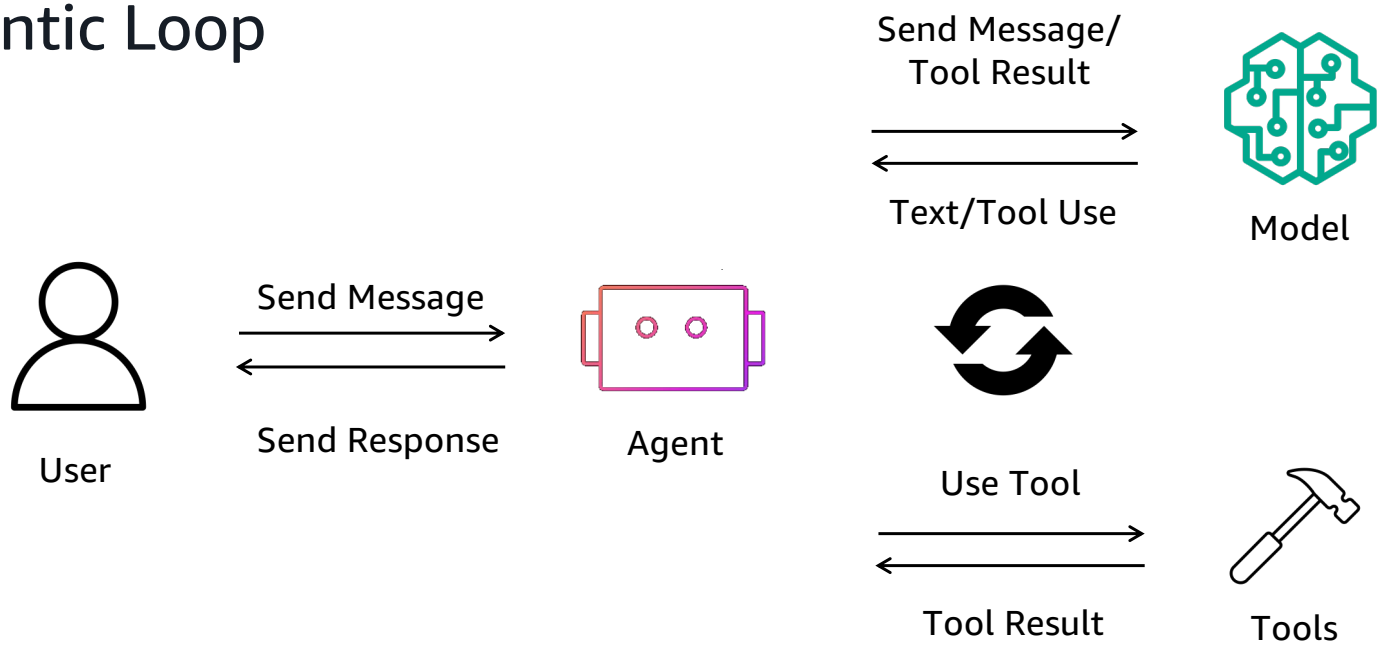


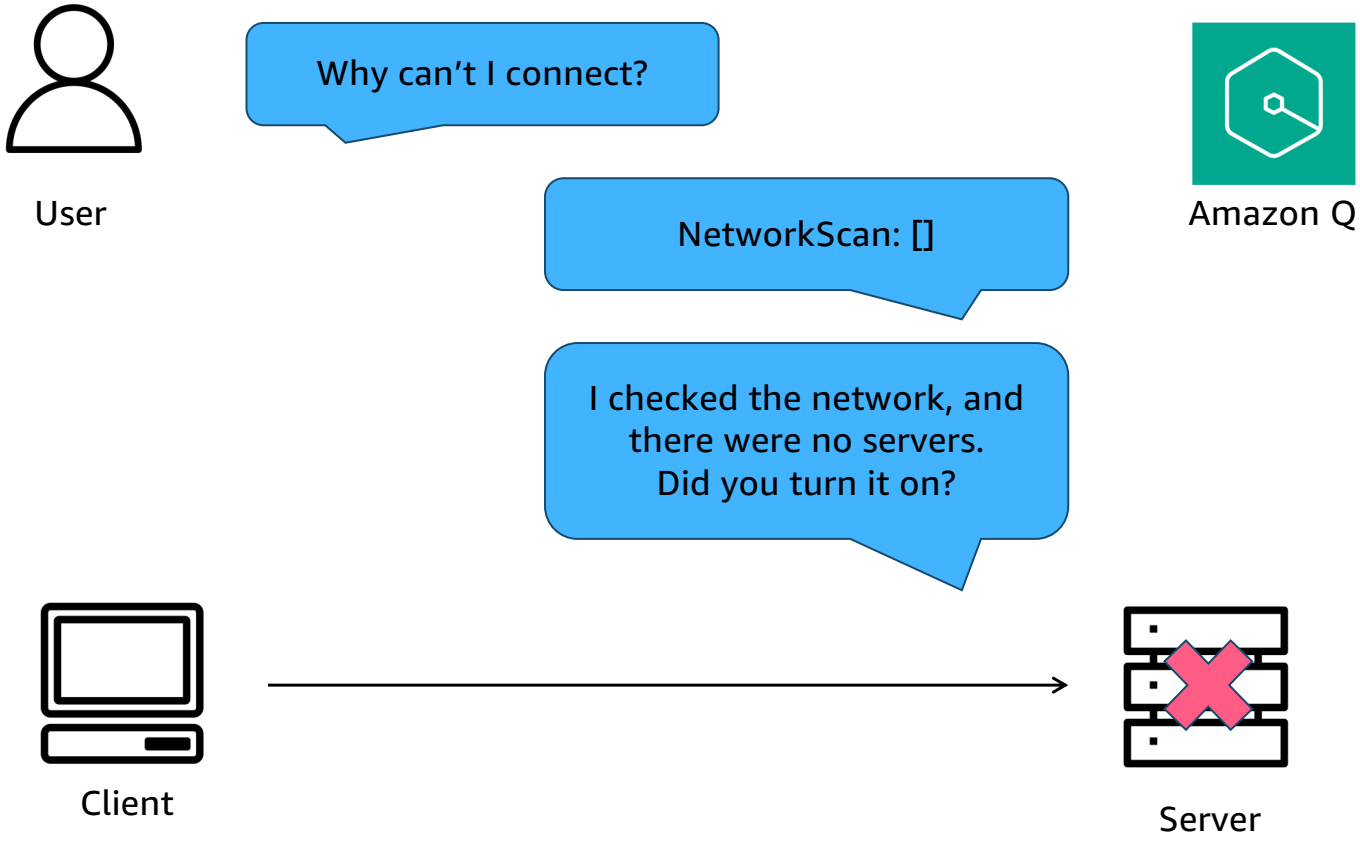
The Model-Driven Approach

The Model-Driven Approach

- Let the Agent decide
- Provide flexible tools
- Informative error messages

Agentic Loop

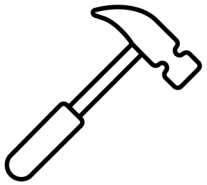




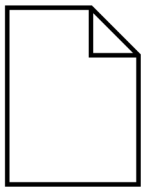
Your First Strands Agent

Building Blocks of an Agent

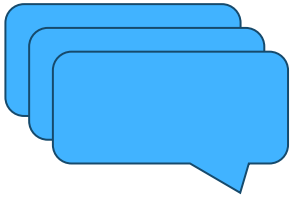
Model



Tools



System Prompt



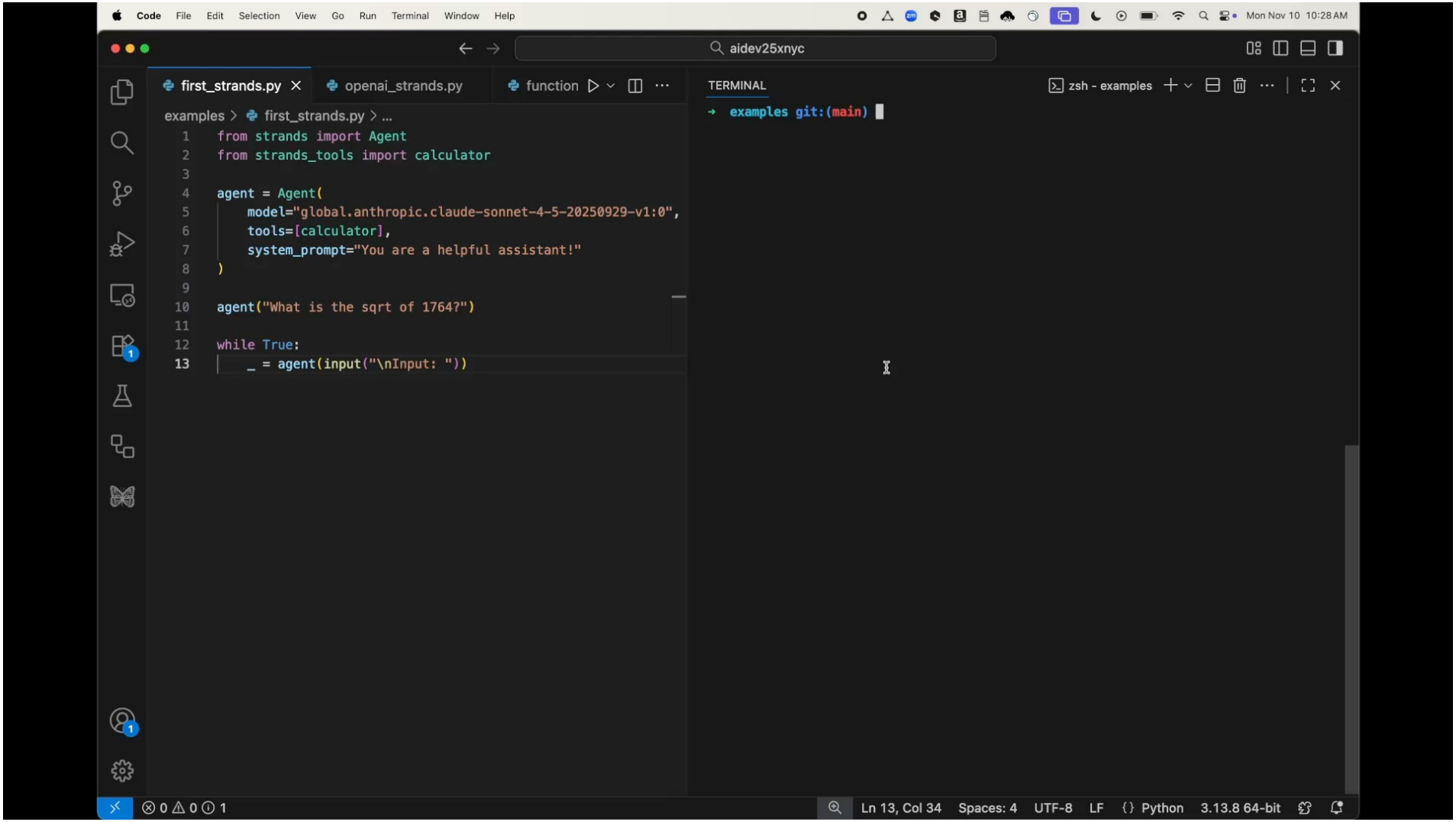
Context

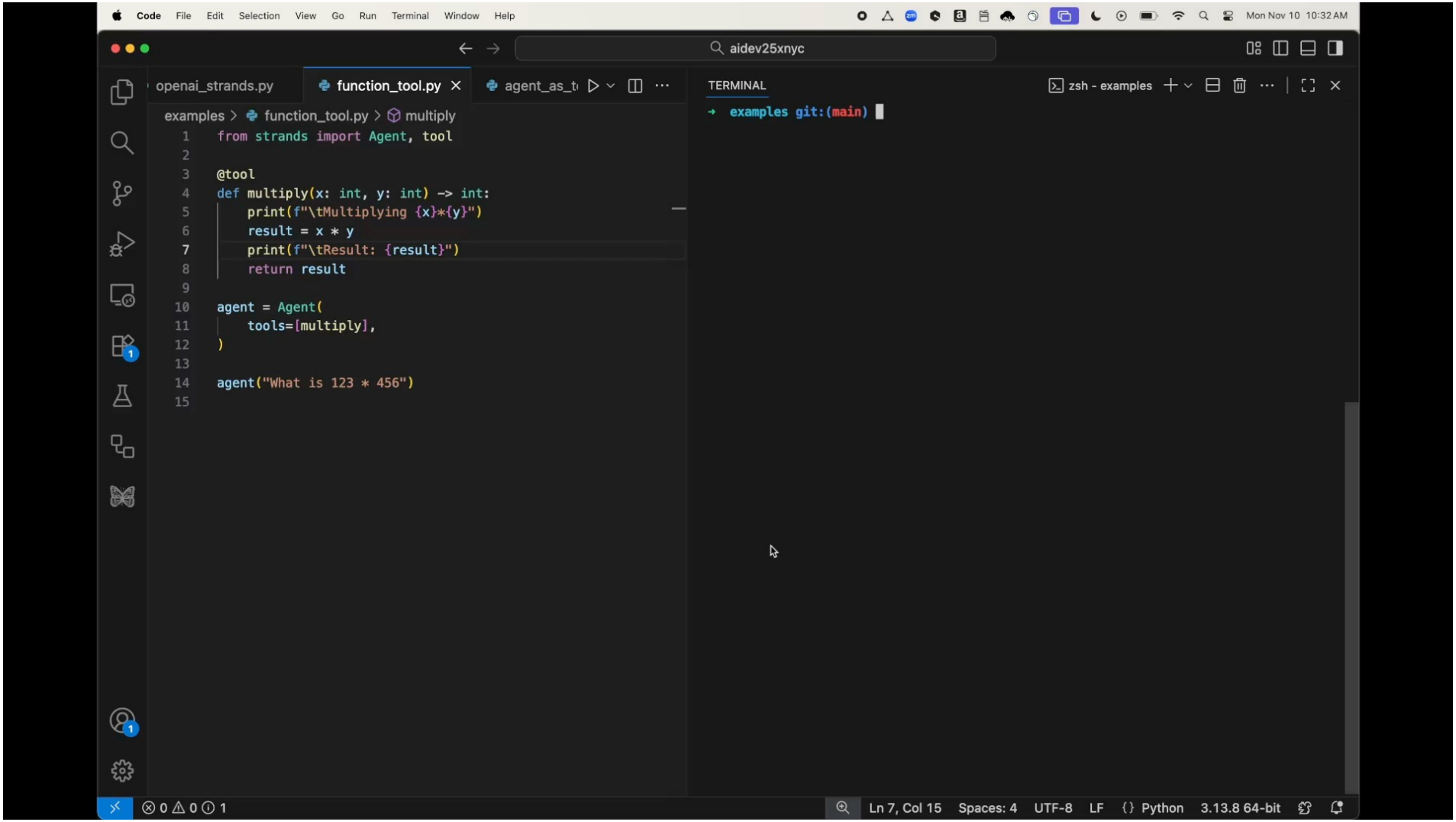
Writing a Strands Agent

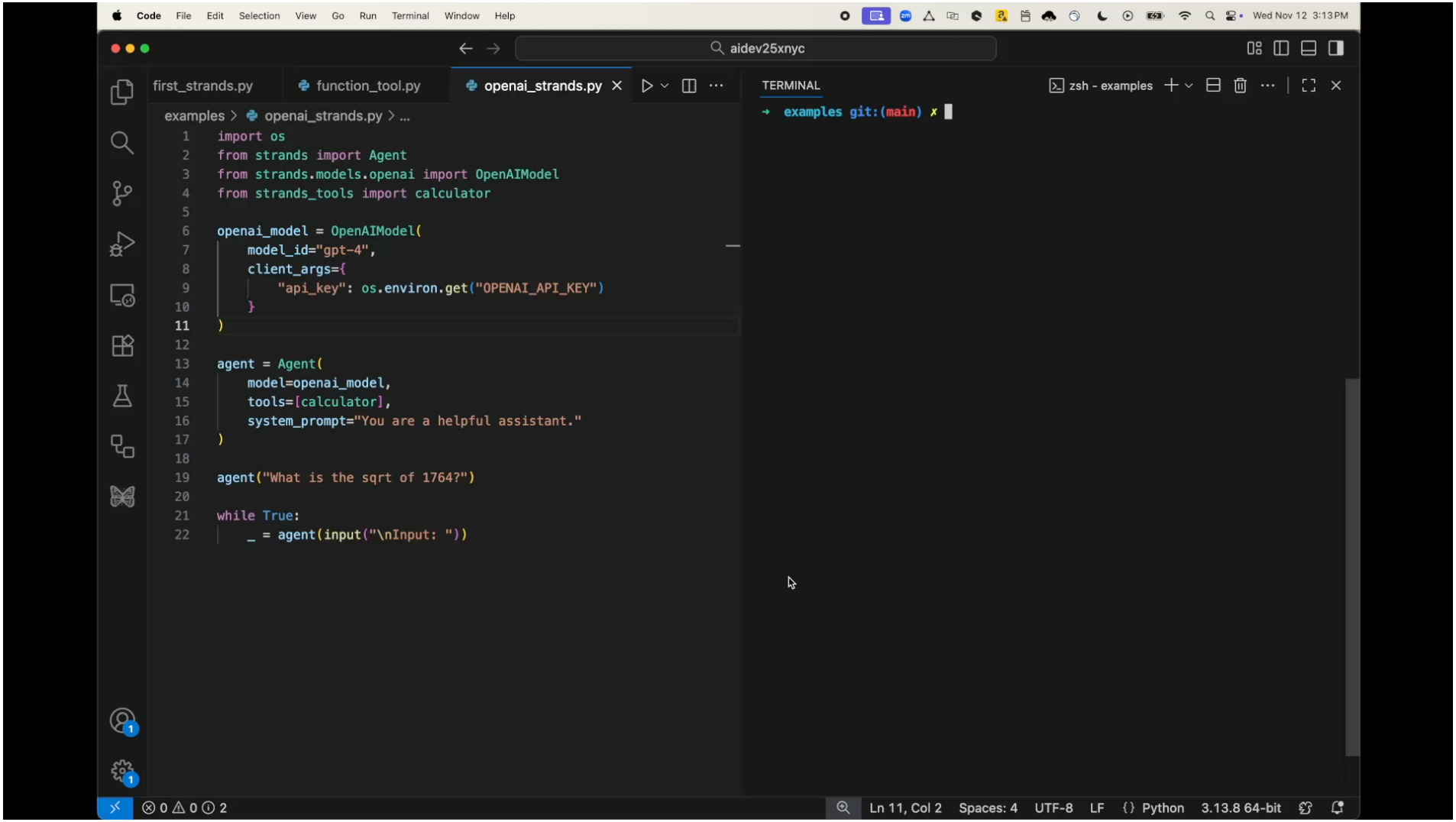
```python
from strands import Agent
from strands_tools import calculator


agent = Agent(
  model="global.anthropic.claude-sonnet-4-5-20250929-v1:0",
  tools=[calculator],
  system_prompt="You are a helpful assistant who talks like a pirate!"
)


agent("What is the sqrt of 1764?")
```

Write Your Own Tools

```python
from strands import Agent, tool

@tool
def multiply(x: int, y: int) -> int:
  return x * y

agent = Agent(
  tools=[multiply],
)

agent("What is 123 * 456")
```

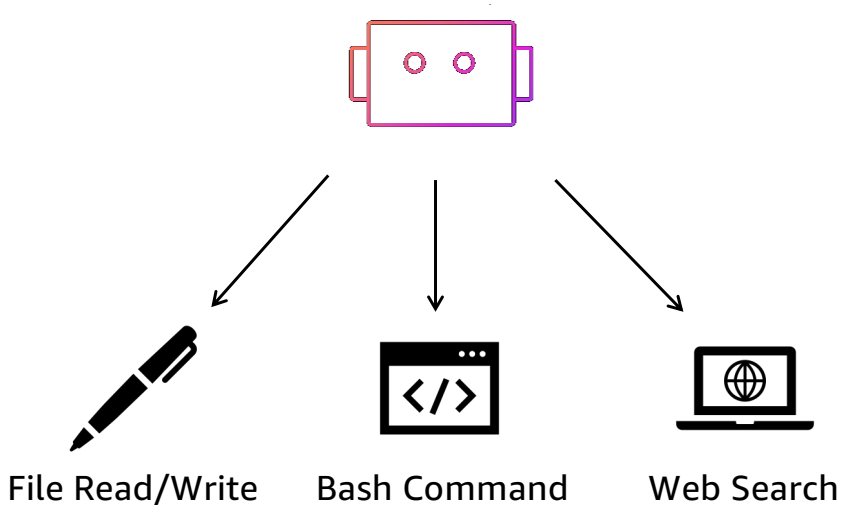

Code File Edit Selection View Go Run Terminal Window Help
Mon Nov 10 10:28 AM
aidev25xnyc
first_strands.py
openai_strands.py
function
examples > first_strands.py > ...
1 from strands import Agent
2 from strands_tools import calculator
3
4 agent = Agent(
5 model="global.anthropic.claude-sonnet-4-5-20250929-v1:0",
6 tools=[calculator],
7 system_prompt="You are a helpful assistant!"
8)
9
10 agent("What is the sqrt of 1764?")
11
12 while True:
13 _ = agent(input("\nInput: "))
TERMINAL
zsh - examples
examples git:(main)
Ln 13, Col 34 Spaces: 4 UTF-8 LF Python 3.13.8 64-bit


Code File Edit Selection View Go Run Terminal Window Help
Mon Nov 10 10:32 AM
aidev25xnyc
openai_strands.py
function_tool.py
agent_as_t
examples > function_tool.py > multiply
1 from strands import Agent, tool
2
3 @tool
4 def multiply(x: int, y: int) -> int:
5 print(f"\tMultiplying {x}*{y}")
6 result = x * y
7 print(f"\tResult: {result}")
8 return result
9
10 agent = Agent(
11 tools=[multiply],
12)
13
14 agent("What is 123 * 456")
15
TERMINAL
zsh - examples
examples git:(main)
Ln 7, Col 15 Spaces: 4 UTF-8 LF Python 3.13.8 64-bit

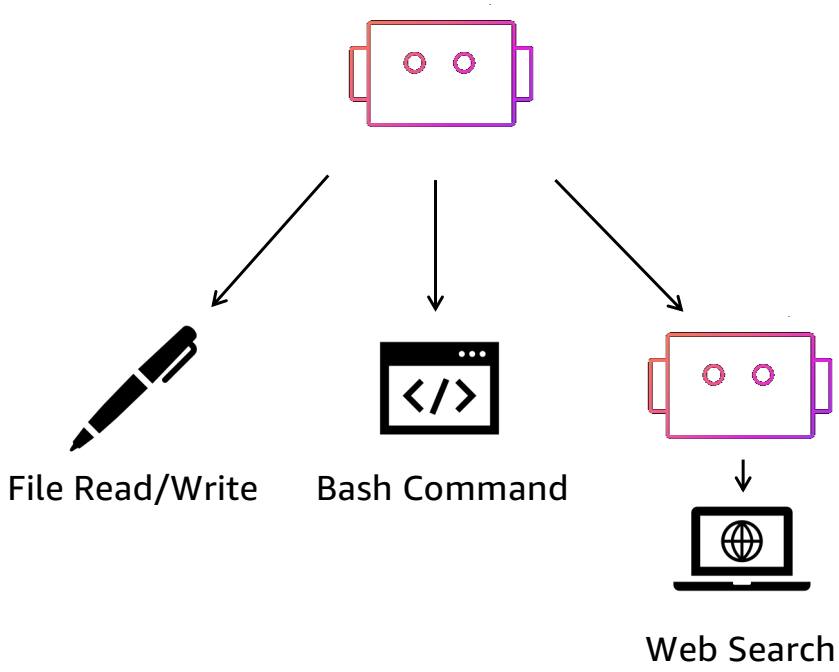

Code File Edit Selection View Go Run Terminal Window Help
Wed Nov 12 3:13 PM
aidev25xnyc
first_strands.py
function_tool.py
openai_strands.py
examples > openai_strands.py > ...
import os
from strands import Agent
from strands.models.openai import OpenAIModel
from strands_tools import calculator
openai_model = OpenAIModel(
model_id="gpt-4",
client_args={
"api_key": os.environ.get("OPENAI_API_KEY")
}
)
agent = Agent(
model=openai_model,
tools=[calculator],
system_prompt="You are a helpful assistant."
)
agent("What is the sqrt of 1764?")
while True:
_ = agent(input("\nInput: "))
TERMINAL
zsh - examples
examples git:(main) ✗
Ln 11, Col 2
Spaces: 4
UTF-8
LF
Python
3.13.8 64-bit

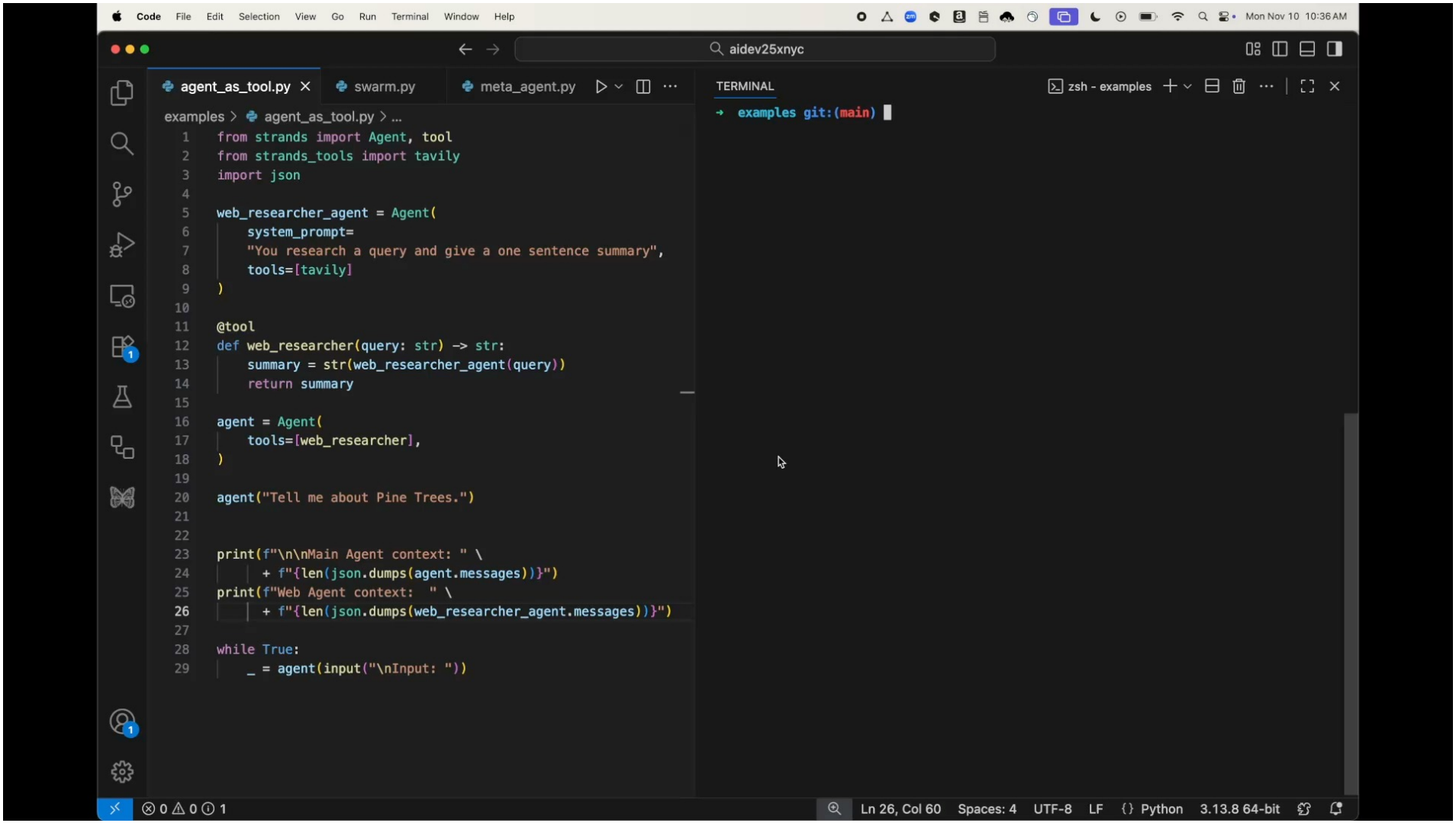
Multi and Meta Agents

Basic Agent

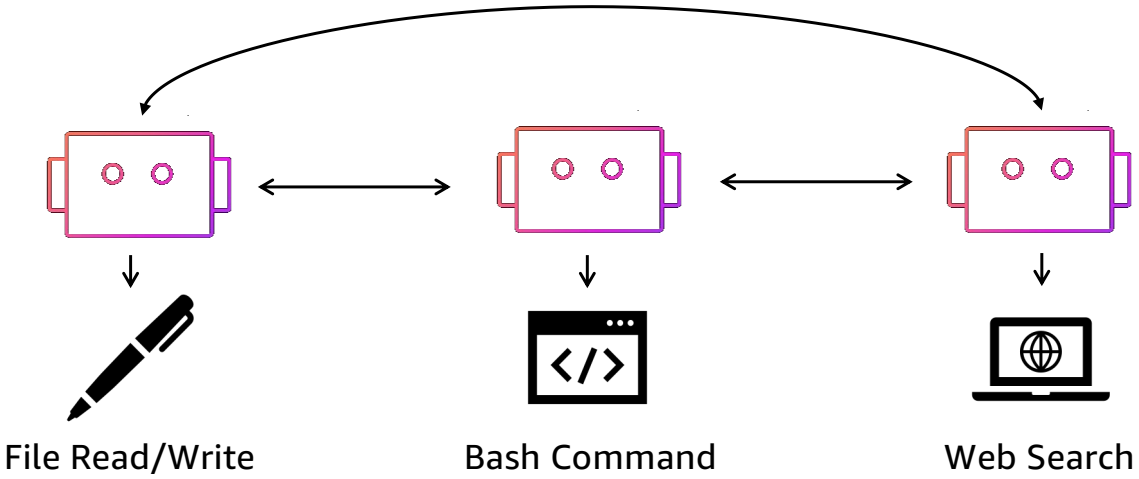


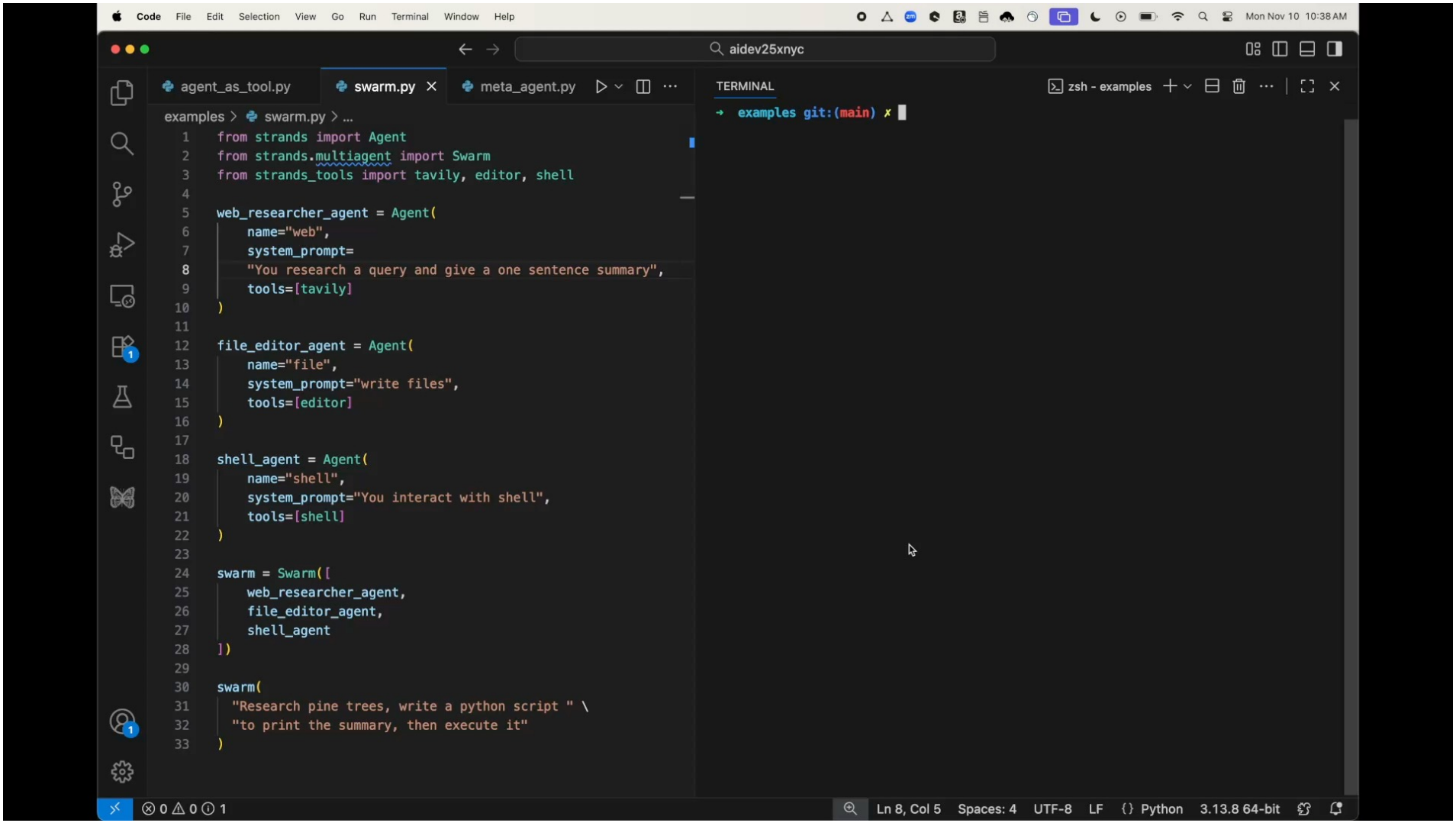
Agents as Tools



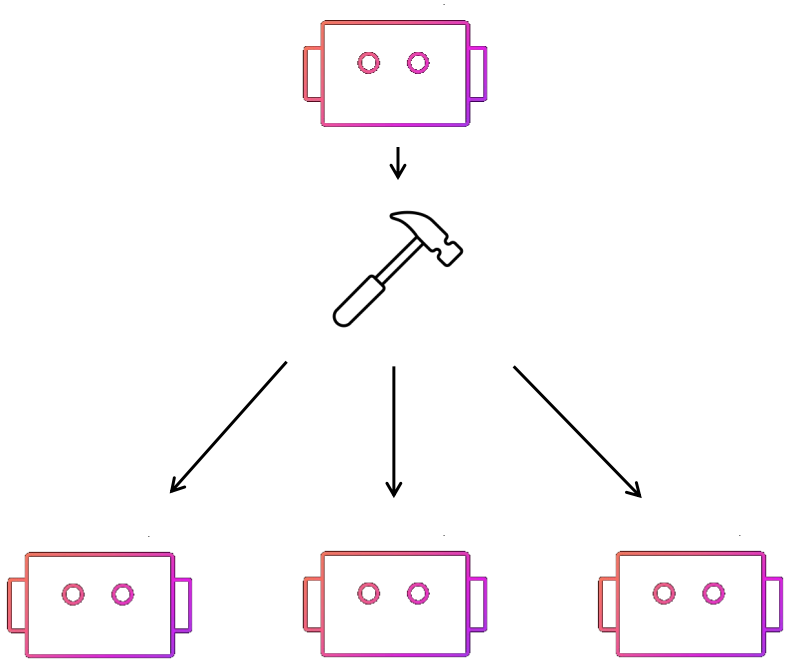

Code File Edit Selection View Go Run Terminal Window Help
Mon Nov 10 10:36 AM
aidev25xnyc
agent_as_tool.py
swarm.py
meta_agent.py
examples > agent_as_tool.py > ...
from strands import Agent, tool
from strands_tools import tavily
import json
web_researcher_agent = Agent(
system_prompt=
"You research a query and give a one sentence summary",
tools=[tavily]
)
@tool
def web_researcher(query: str) -> str:
summary = str(web_researcher_agent(query))
return summary
agent = Agent(
tools=[web_researcher],
)
agent("Tell me about Pine Trees.")
print(f"\n\nMain Agent context: " \
+ f"{len(json.dumps(agent.messages))}")
print(f"Web Agent context: " \
+ f"{len(json.dumps(web_researcher_agent.messages))}")
while True:
_ = agent(input("\nInput: "))
TERMINAL
zsh - examples
examples git:(main)
0 0 1
Ln 26, Col 60
Spaces: 4
UTF-8
LF
Python
3.13.8 64-bit

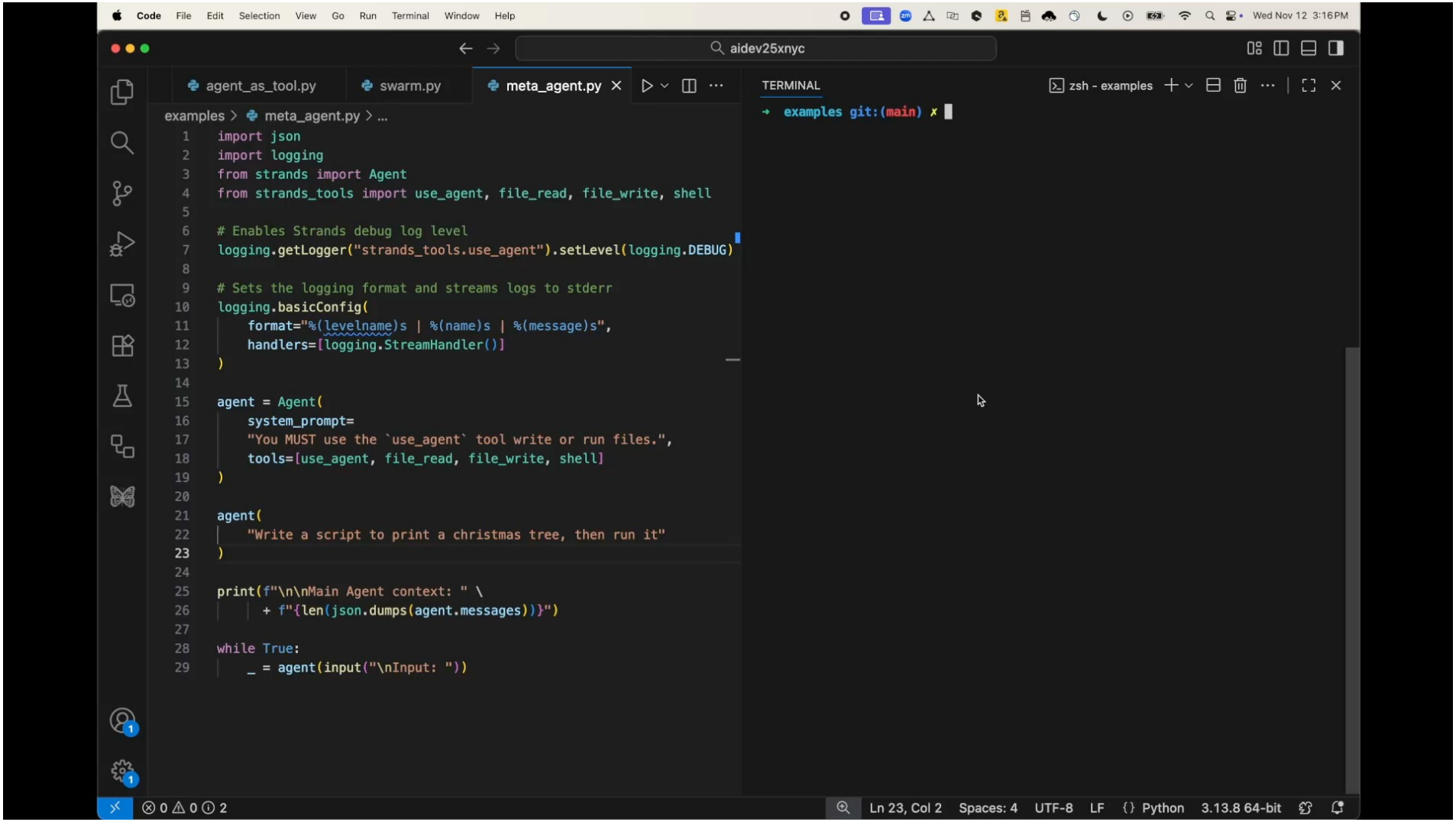
Swarms




Code File Edit Selection View Go Run Terminal Window Help
Mon Nov 10 10:38 AM
aidev25xnyc
agent_as_tool.py
swarm.py
meta_agent.py
examples > swarm.py > ...
from strands import Agent
from strands.multiagent import Swarm
from strands_tools import tavily, editor, shell
web_researcher_agent = Agent(
name="web",
system_prompt=
"You research a query and give a one sentence summary",
tools=[tavily]
)
file_editor_agent = Agent(
name="file",
system_prompt="write files",
tools=[editor]
)
shell_agent = Agent(
name="shell",
system_prompt="You interact with shell",
tools=[shell]
)
swarm = Swarm([
web_researcher_agent,
file_editor_agent,
shell_agent
])
swarm(
"Research pine trees, write a python script " \
"to print the summary, then execute it"
)
TERMINAL
zsh - examples
examples git:(main) ✗
Ln 8, Col 5 Spaces: 4 UTF-8 LF Python 3.13.8 64-bit

Meta Agents



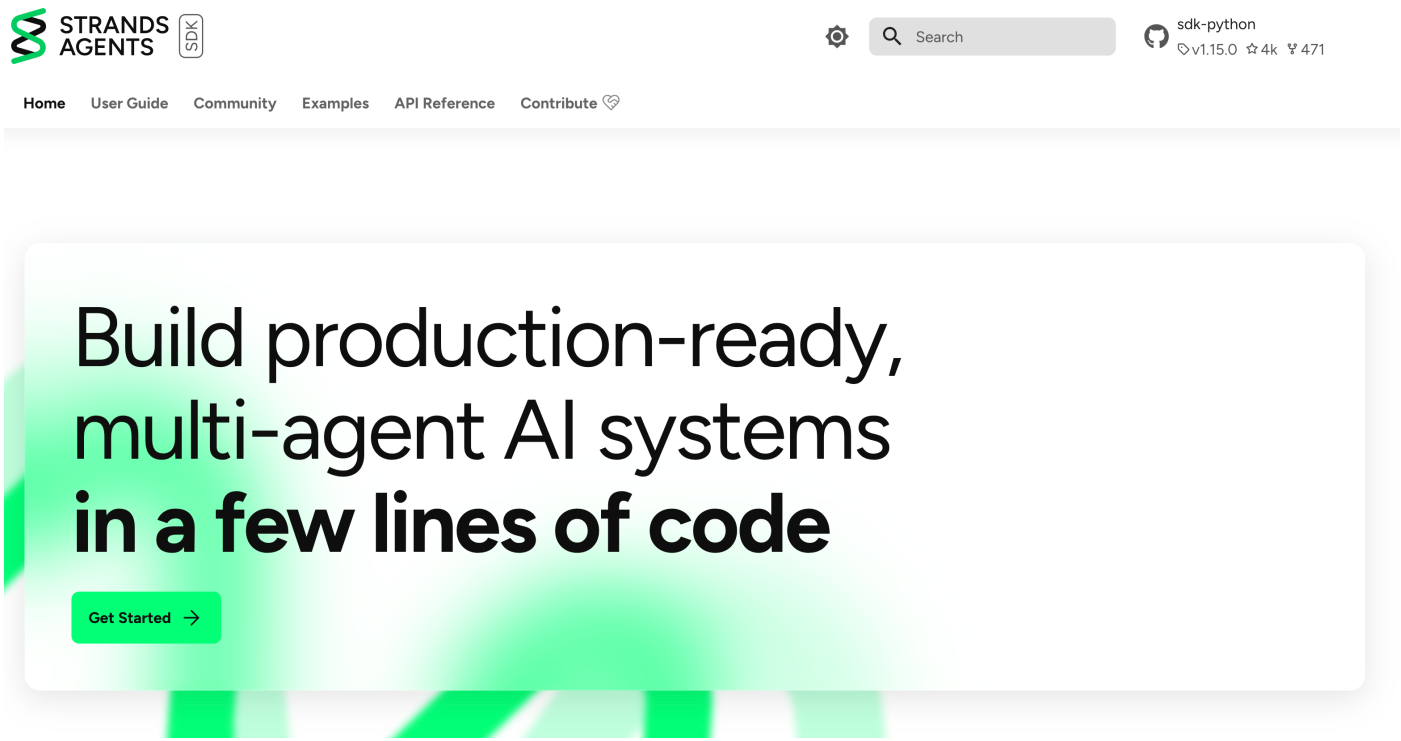

Code File Edit Selection View Go Run Terminal Window Help
Wed Nov 12 3:16 PM
aidev25xnyc
agent_as_tool.py
swarm.py
meta_agent.py
examples > meta_agent.py > ...
import json
import logging
from strands import Agent
from strands_tools import use_agent, file_read, file_write, shell
# Enables Strands debug log level
logging.getLogger("strands_tools.use_agent").setLevel(logging.DEBUG)
# Sets the logging format and streams logs to stderr
logging.basicConfig(
format="%(levelname)s | %(name)s | %(message)s",
handlers=[logging.StreamHandler()]
)
agent = Agent(
system_prompt=
"You MUST use the `use_agent` tool write or run files.",
tools=[use_agent, file_read, file_write, shell]
)
agent(
"Write a script to print a christmas tree, then run it"
)
print(f"\n\nMain Agent context: " \
+ f"{len(json.dumps(agent.messages))}")
while True:
_ = agent(input("\nInput: "))
TERMINAL
zsh - examples
examples git:(main) ✗
Ln 23, Col 2 Spaces: 4 UTF-8 LF Python 3.13.8 64-bit

Takeaways

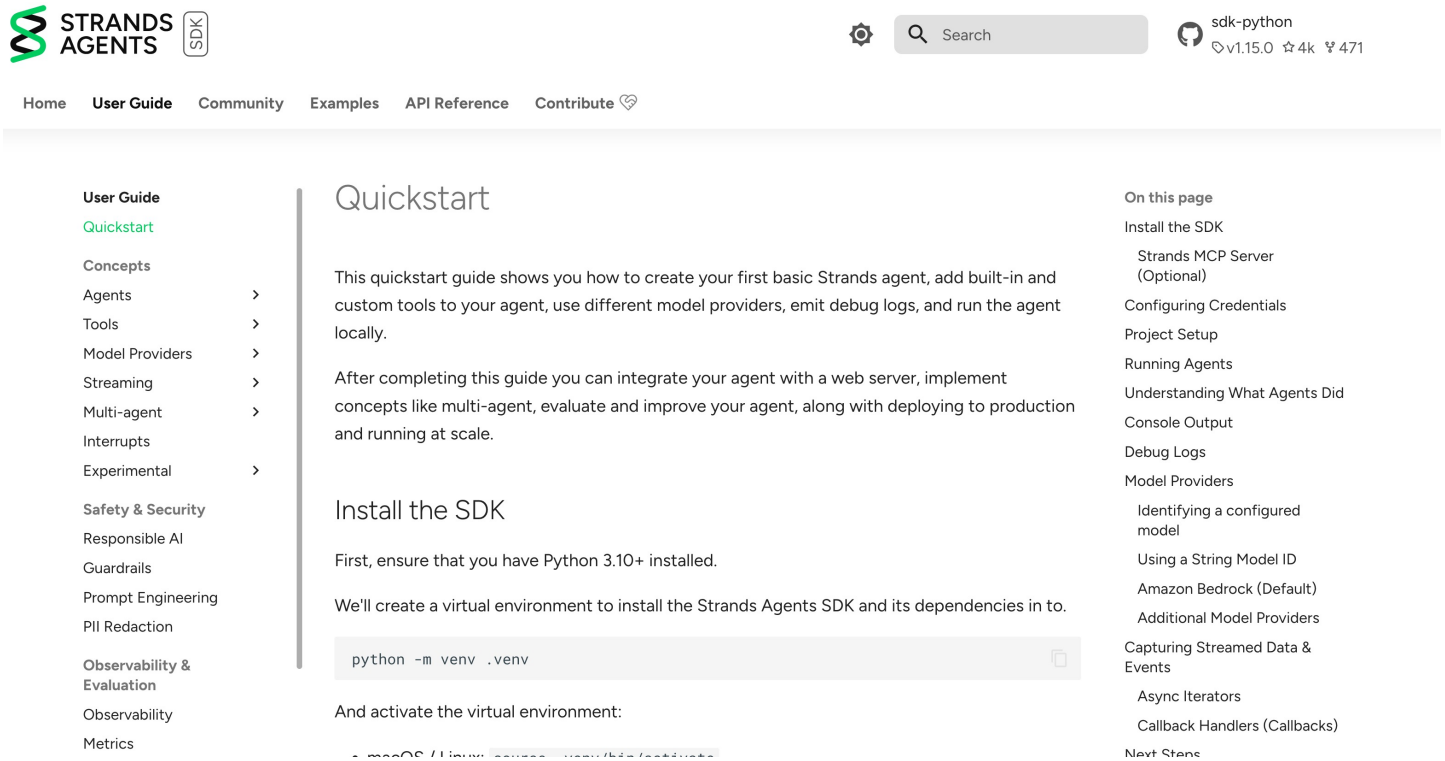
Design Model-Driven Agents

- Put the Agent in control
- Informative Errors
- Flexible Tool
- Think about context bloat

Start building a Strands Agent! – strandsagents.com



Easy Getting Started Guide



Thank you!

Nicholas Clegg
https://www.linkedin.com/in/nicholasclegg