맞다 AI가

Microsoft Foundry로 오픈클로 멀티에이전트 시스템 구축해 보기

황지현 – Microsoft Student Ambassadors Senior

황지현

Microsoft Student Ambassadors- Senior

경북대학교

@zihyeon08

@jihyeon081


Microsoft


OceanLightAI


AIFactory

Claw and Agent Harness in Microsoft Foundry

Shawn Henry, Principal Group PM, Microsoft Foundry

Amanda Foster, Product Manager, Microsoft Foundry

Glenn Condron, Principal PM, Microsoft Foundry

오늘의 핵심 질문

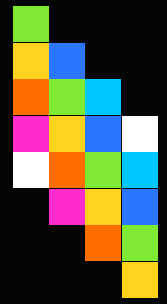
에이전트를 실제 업무 환경에서 오래, 안전하게 운영할 수 있을까?

1. 상태는 어디에 저장되는가?

2. 도구와 권한은 어떻게 통제하는가?

3. 사람은 언제 개입하는가?

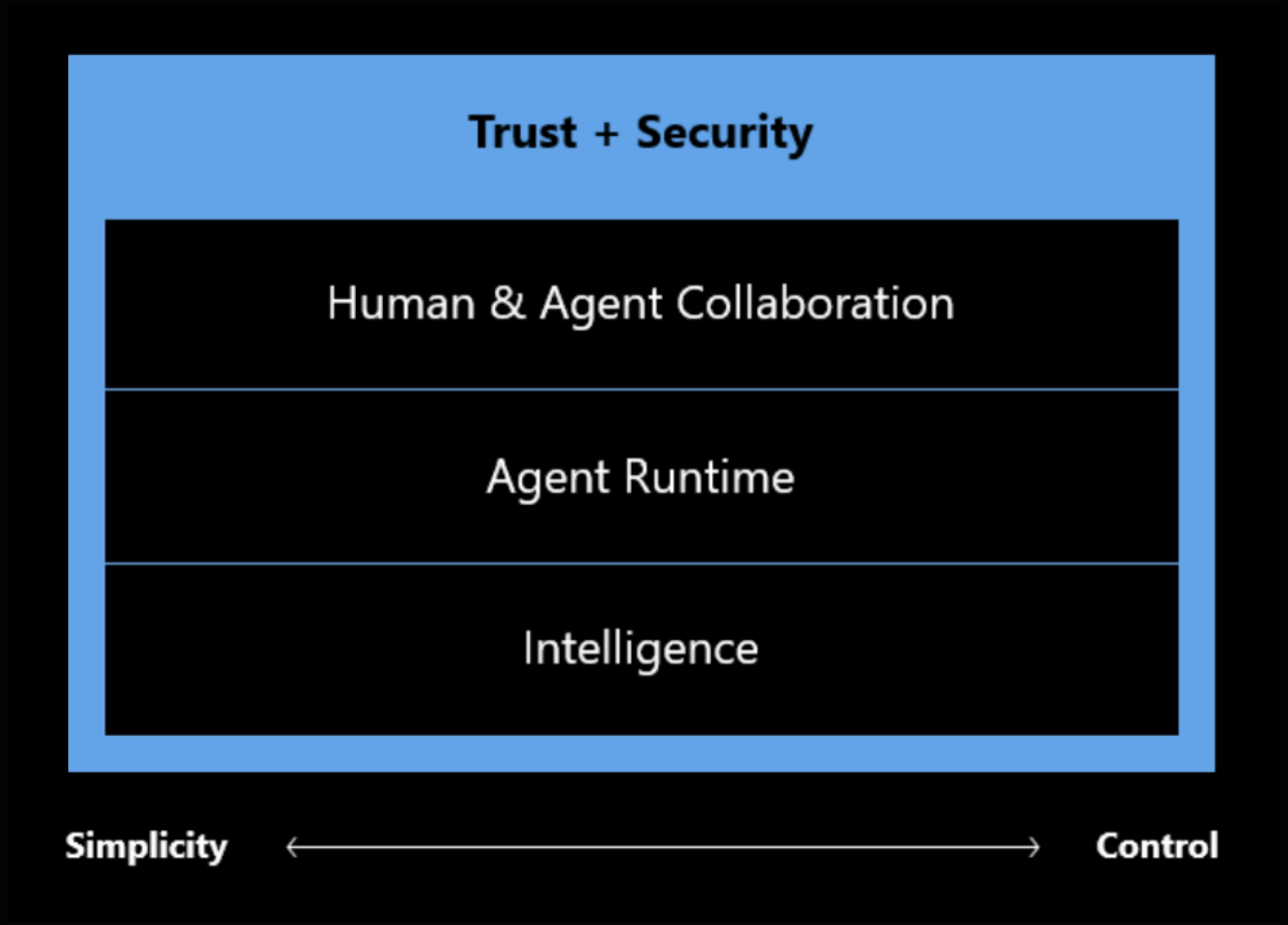
4. Teams / Copilot으로 어떻게 배포하는가?



전체 흐름

개념 설명 → 데모 3개 → 실제 업무 채널 배포

Foundry의 3개 계층



Foundry가 제공하는 것

- 코드·프롬프트·스킬·도구 정의
- Foundry에서 에이전트를 실행하고 호스팅
- Teams, M365 Copilot으로 사용자에게 배포

최근 6개월의 변화

코딩 에이전트가 강력해지고, Claw-style 패턴이 등장했다

개인 개발 도구에서 → 엔터프라이즈 에이전트 운영 패턴으로

Agent Harness란?

모델을 실제로 "일하는 에이전트"로 만드는 실행 장치

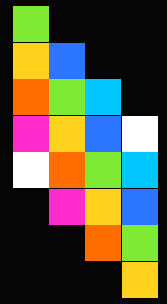
Agent Harness 정의

에이전트가 더 길고 복잡한 작업을 수행하도록 감싸는 도구 세트

Agent
= Engine

+ 바퀴, 핸들, 좌석, 브레이크

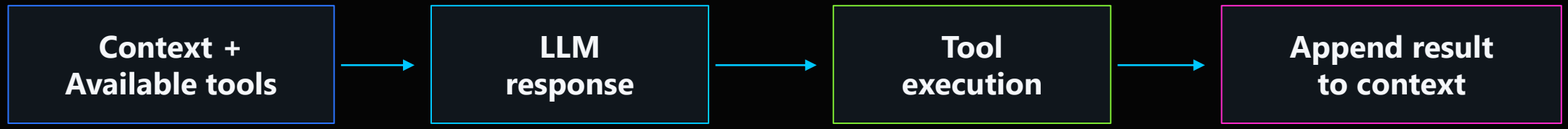
Harness
= Car



엔진만으로는 달릴 수 없다.

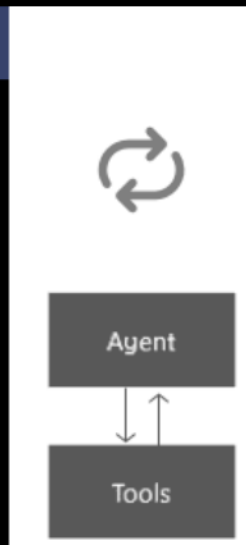
도구, 메모리, 권한, 환경, 사람의 승인이 필요하다.

Agent Loop

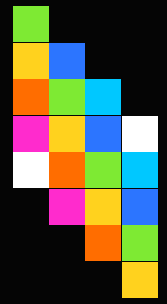


Agent Loop

- Context Management
- Skills and Tools
- Sub-Agents
- Memory and Session Persistence
- Lifecycle Hooks
- Permissions and HITL

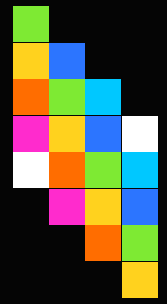



```
while true:
    response = send_to_llm(context,
available_tools)
    if response.contains_tool_calls:
        execute each tool
        append results to context
        continue
if response.is_done:
    break
```



Harness의 6개 구성요소

Context Management	Skills & Tools	Sub-agents
Memory & Sessions	Lifecycle Hooks	Permissions & HITL



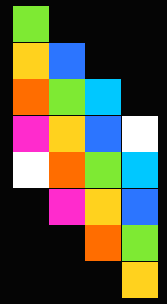
이 6개가 있어야 에이전트를 “프로덕션 단위”로 다룰 수 있다.

Context Management

컨텍스트는 쌓이고, 하네스는 압축한다

Memory | Messages | Tool calls | Reasoning | Skills → Context Compaction

오래 실행되는 에이전트일수록 context window 관리가 중요하다



Skills and Tools

MCP

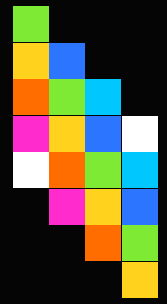
OpenAPI

File system

Code

Shell

Skills .md

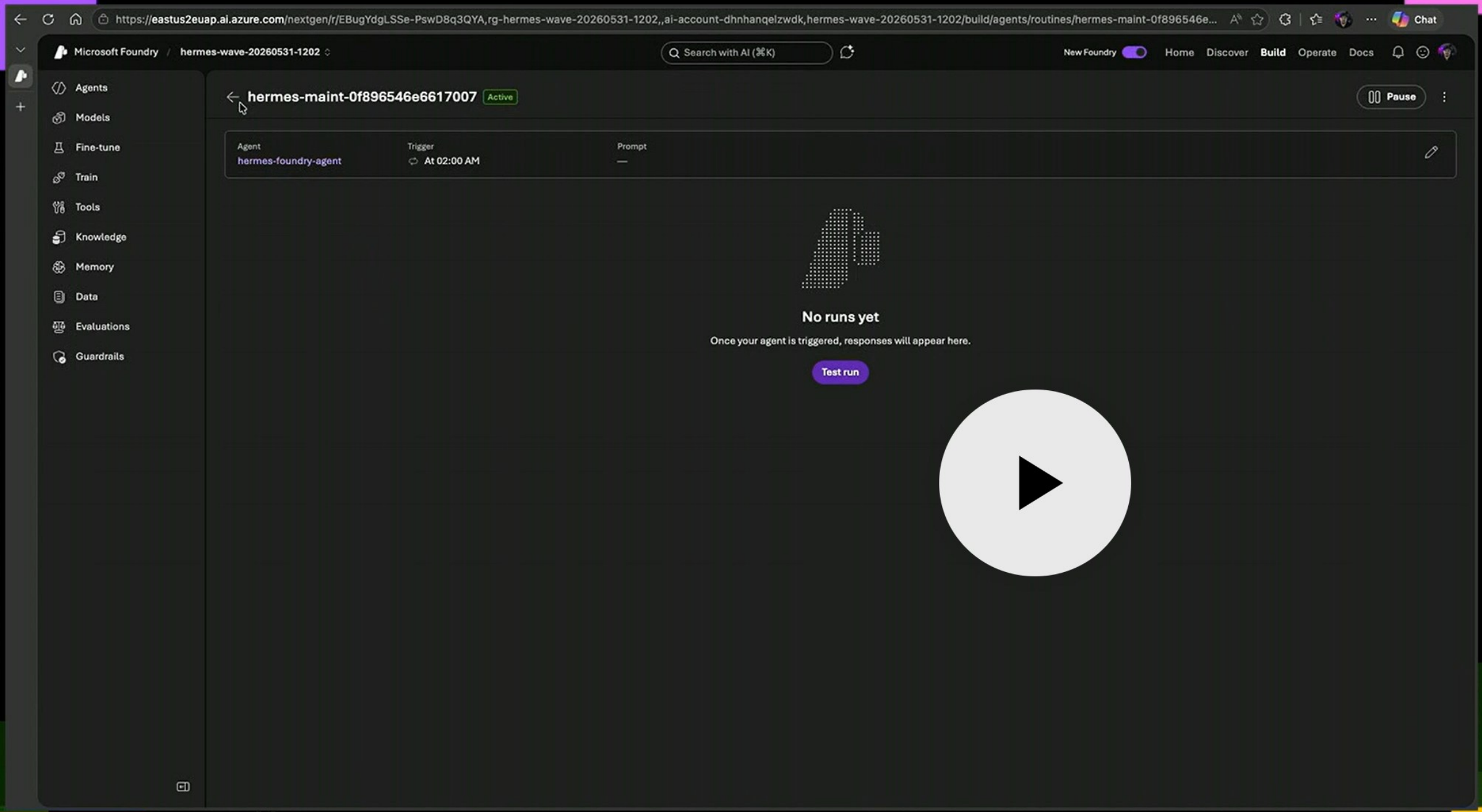


단순 응답 생성 → 파일 읽기/쓰기, 코드 실행, 시스템 변경 수행

Permissions & HITL

강력한 에이전트일수록 인간의 승인 지점이 필요하다

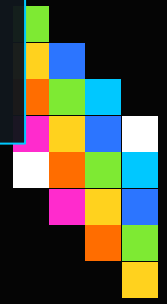
- 파일 삭제, 코드 실행, 외부 시스템 변경
- Teams / Outlook 같은 채널에서 승인 가능
- 정책과 권한을 하네스 레이어에서 통제


https://eastus2euap.ai.azure.com/nextgen/r/EBugYdgLSSe-PswD8q3QYA,rg-hermes-wave-20260531-1202,,ai-account-dhnhanqelzwdk,hermes-wave-20260531-1202/build/agents/routines/hermes-maint-0f896546e...
Chat
Microsoft Foundry
hermes-wave-20260531-1202
Search with AI (⌘K)
New Foundry
Home
Discover
Build
Operate
Docs
Agents
Models
Fine-tune
Train
Tools
Knowledge
Memory
Data
Evaluations
Guardrails
hermes-maint-0f896546e6617007
Active
Pause
Agent
hermes-foundry-agent
Trigger
At 02:00 AM
Prompt
—
No runs yet
Once your agent is triggered, responses will appear here.
Test run

Hermes 데모 해설

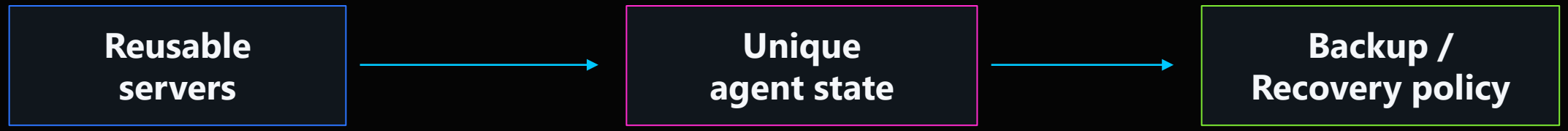
- 로컬 TUI는 내 컴퓨터, backend는 Foundry Hosted Agent
- session ID마다 고유 sandbox / file system 생성
- routines가 maintenance와 backup 역할 수행
- idle 상태에서는 꺼졌다가 필요 시 다시 작동

Claw-style agent는 오래 일할수록 skill, memory, file state를 축적한다



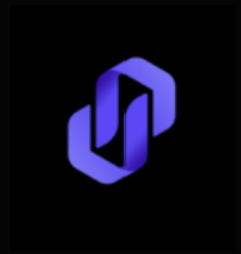
Agent State의 문제

"고유해지는 에이전트"를 어떻게 운영할 것인가?

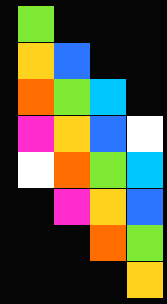


- 에이전트가 잘할수록 더 개인화되고 고유해짐
- 그 상태가 장애 복구·비용·보안 이슈가 됨
- Foundry Hosted Agents의 sandbox와 routine이 한 가지 해법

Microsoft Agent Framework



기성 하네스가 아니라 직접 custom harness를 만드는 방법

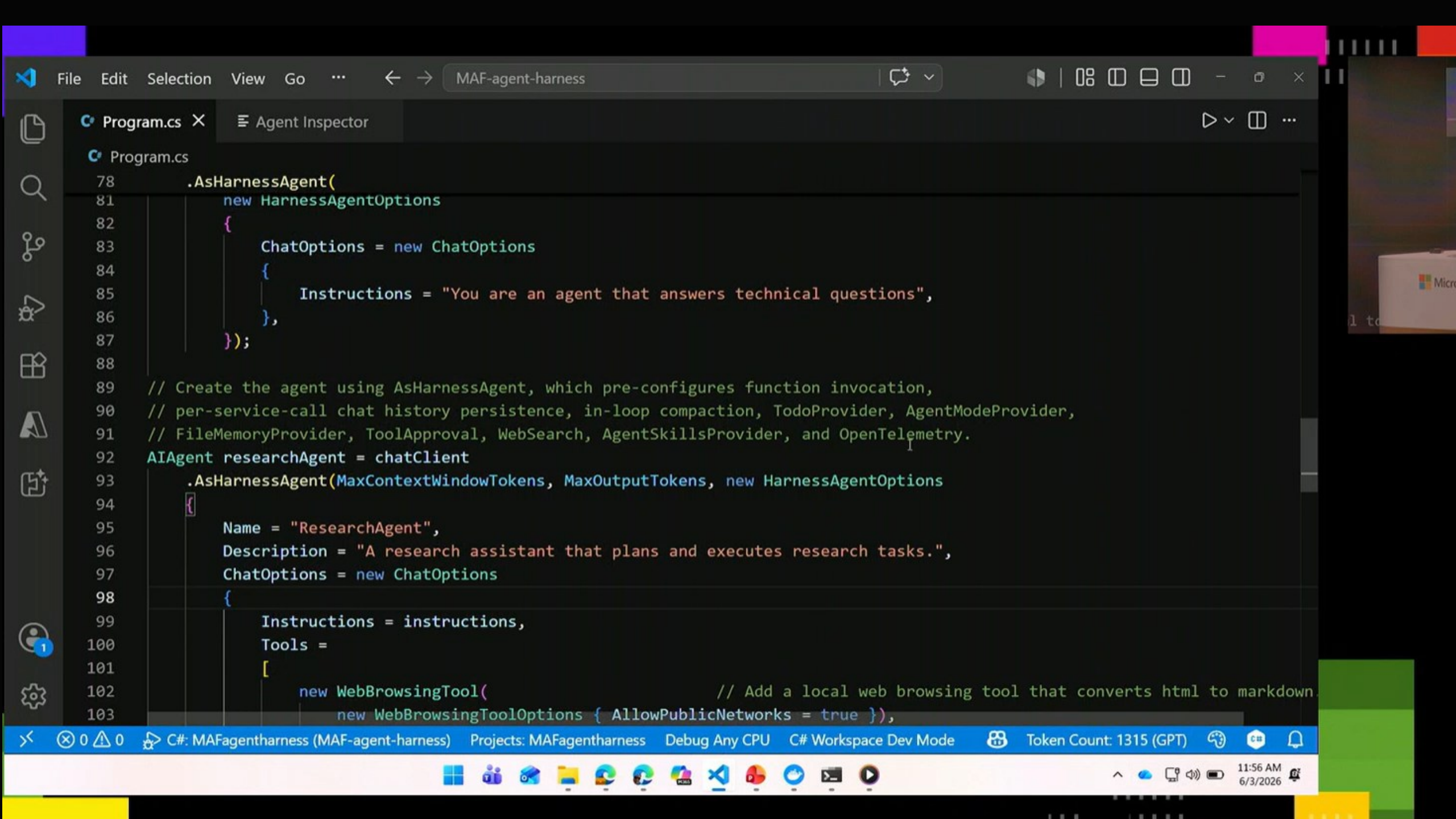


Harness

Deep research | **Content generation** | **Custom Harnesses**

Coding | **Goal-driven agents**

Common Tools	Context	Planning	Middleware
File system	Prompts	Todo	Context compaction
Code execution	Skills	Subagents	Tool selection
Shell execution	Memory		Permissions


File Edit Selection View Go ···
MAF-agent-harness
Program.cs
Agent Inspector
Program.cs
78 .AsHarnessAgent(
81 new HarnessAgentOptions
82 {
83 ChatOptions = new ChatOptions
84 {
85 Instructions = "You are an agent that answers technical questions",
86 },
87 });
88
89 // Create the agent using AsHarnessAgent, which pre-configures function invocation,
90 // per-service-call chat history persistence, in-loop compaction, TodoProvider, AgentModeProvider,
91 // FileMemoryProvider, ToolApproval, WebSearch, AgentSkillsProvider, and OpenTelemetry.
92 AIAgent researchAgent = chatClient
93 .AsHarnessAgent(MaxContextWindowTokens, MaxOutputTokens, new HarnessAgentOptions
94 {
95 Name = "ResearchAgent",
96 Description = "A research assistant that plans and executes research tasks.",
97 ChatOptions = new ChatOptions
98 {
99 Instructions = instructions,
100 Tools =
101 [
102 new WebBrowsingTool(// Add a local web browsing tool that converts html to markdown
103 new WebBrowsingToolOptions { AllowPublicNetworks = true }),
0 0 C#: MAFagentharness (MAF-agent-harness) Projects: MAFagentharness Debug Any CPU C# Workspace Dev Mode Token Count: 1315 (GPT)
11:56 AM
6/3/2026

MAF 데모 해설

하네스는 Agent Loop에 운영 기능을 붙인다

1 Plan

먼저 계획을 세움

2 Ask

필요하면 사람에게 질문

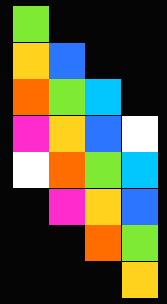
3 Act

도구 호출과 todo로 실행

4 Observe

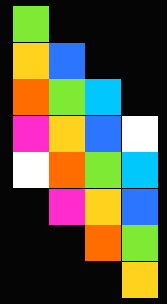
trace와 telemetry로 관찰

에이전트가 단순히 답하는 것이 아니라, 계획하고 승인받고 실행한다



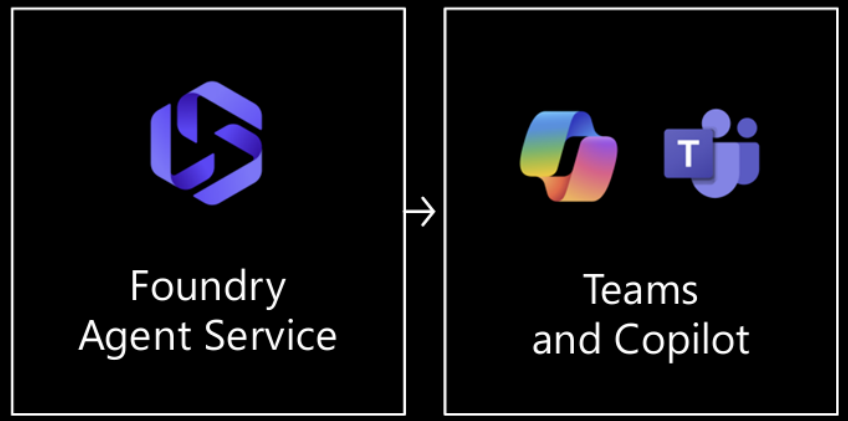
Teams와 Copilot으로 배포

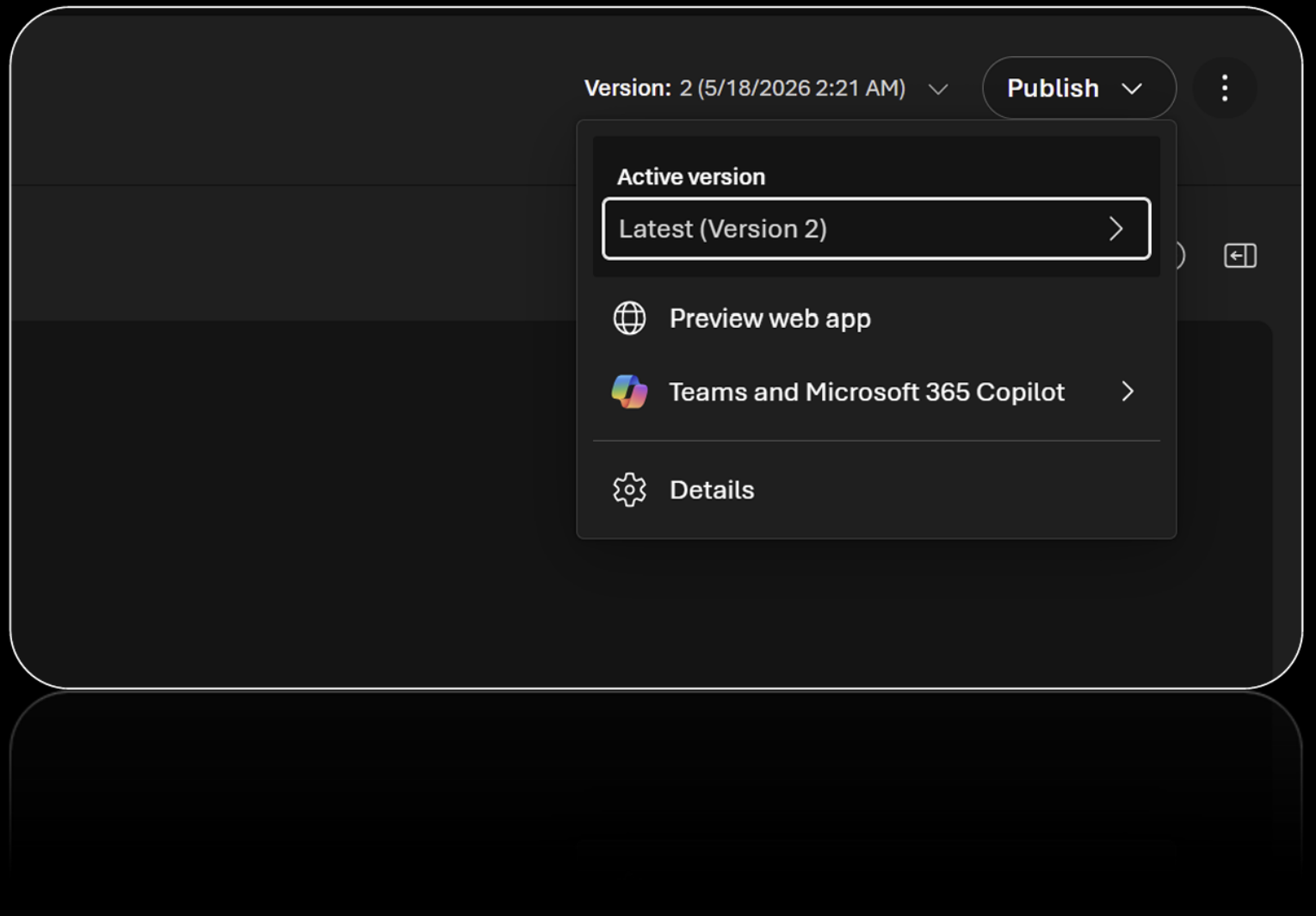
에이전트는 사용자가 일하는 곳에 있어야 한다



Publish to Teams and Microsoft 365 Copilot

Take agents built with Foundry and make them available instantly in M365 Copilot and Teams Chat

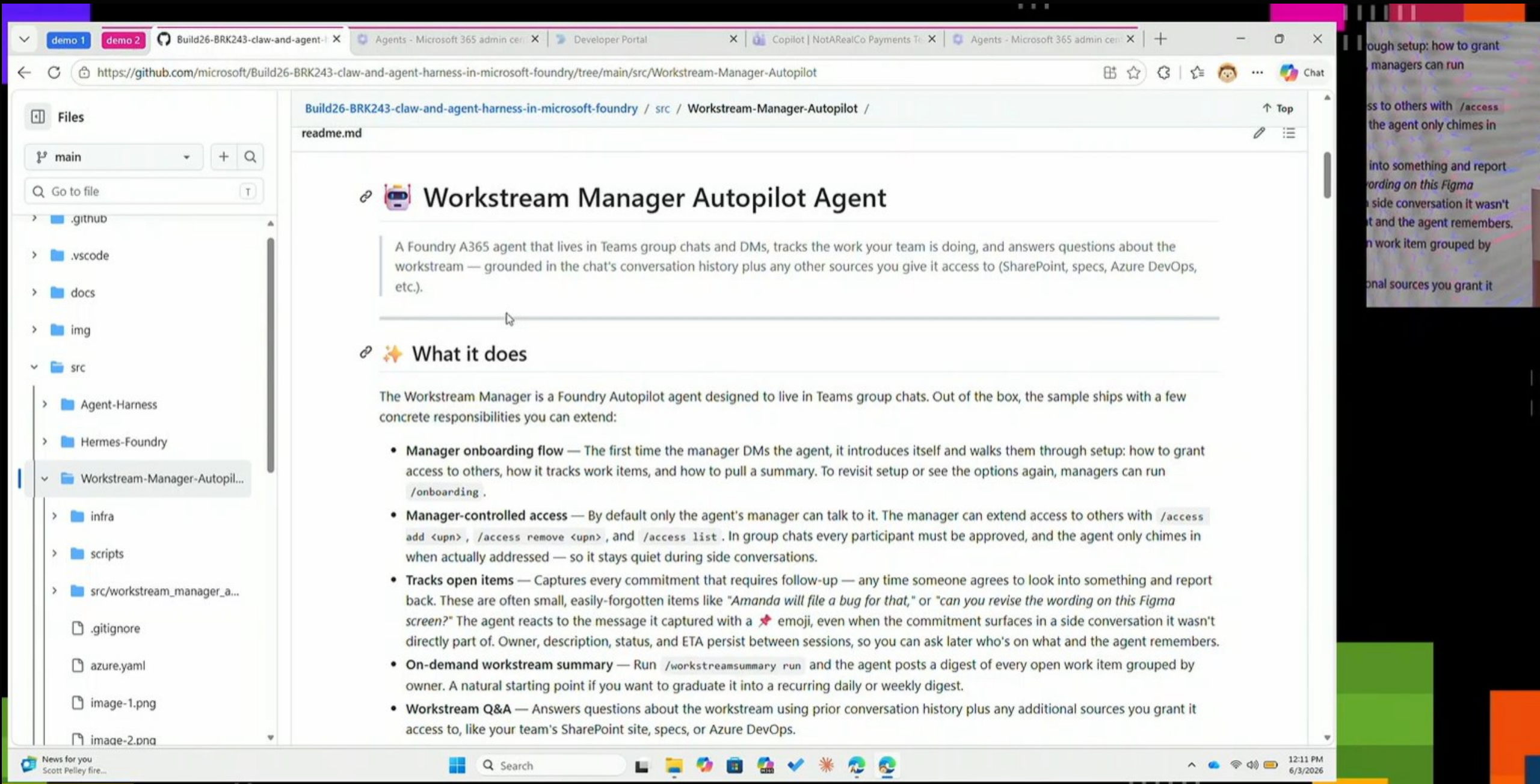




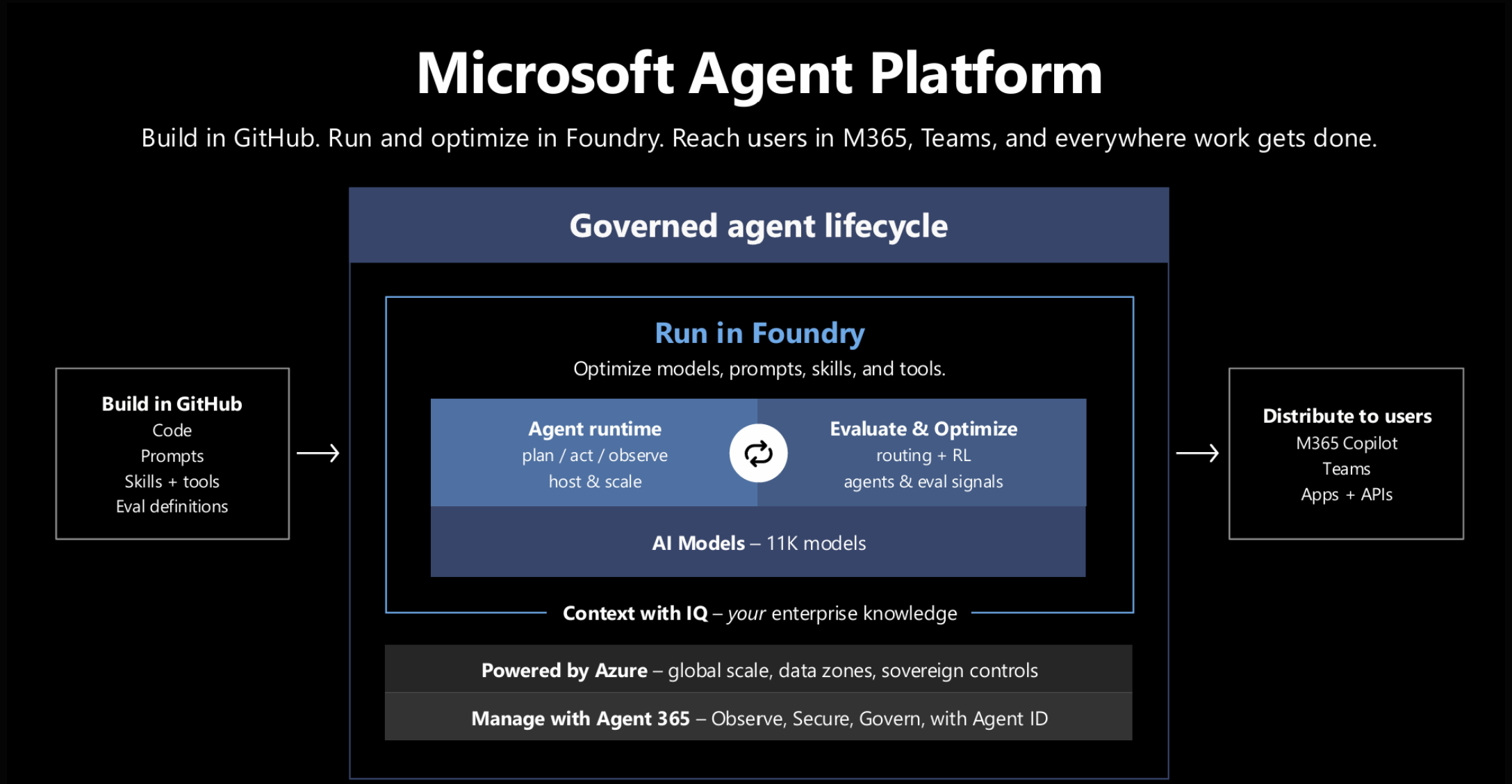
세 가지 에이전트 배포 시 권한

Assistive · Autonomous · Autopilot

		
Assistive On-behalf-of (OBO)	**Autonomous** Headless / no collab surface	**Autopilot agents** Autonomous + collaborative
Acts with your identity, your permissions, your data. Lives inside Copilot or a chat with you. **Example:** Sales Agent in Copilot — drafts a follow-up using your mailbox and CRM access.	Operates independently using its own identity but cannot perform actions that require user accounts. Runs in the background — triggered by events, schedules, or other agents. **Example:** a nightly agent that triages incidents and files tickets.	Operates independently using its own identity AND has a user account, so it gets a mailbox, calendar, OneDrive, etc. Replies inline, asks for input, gets @-mentioned like a person. **Example:** the workstream manager autopilot we ship today.


demo 1
demo 2
Build26-BRK243-claw-and-agent-
Agents - Microsoft 365 admin cen
Developer Portal
Copilot | NotARealCo Payments Te
Agents - Microsoft 365 admin cen
https://github.com/microsoft/Build26-BRK243-claw-and-agent-harness-in-microsoft-foundry/tree/main/src/Workstream-Manager-Autopilot
Chat
Files
main
Go to file
.github
.vscode
docs
img
src
Agent-Harness
Hermes-Foundry
Workstream-Manager-Autopil...
infra
scripts
src/workstream_manager_a...
.gitignore
azure.yaml
image-1.png
image-2.png
Build26-BRK243-claw-and-agent-harness-in-microsoft-foundry / src / Workstream-Manager-Autopilot /
Top
readme.md
Workstream Manager Autopilot Agent
A Foundry A365 agent that lives in Teams group chats and DMs, tracks the work your team is doing, and answers questions about the workstream — grounded in the chat's conversation history plus any other sources you give it access to (SharePoint, specs, Azure DevOps, etc.).
What it does
The Workstream Manager is a Foundry Autopilot agent designed to live in Teams group chats. Out of the box, the sample ships with a few concrete responsibilities you can extend:
Manager onboarding flow — The first time the manager DMs the agent, it introduces itself and walks them through setup: how to grant access to others, how it tracks work items, and how to pull a summary. To revisit setup or see the options again, managers can run /onboarding.
Manager-controlled access — By default only the agent's manager can talk to it. The manager can extend access to others with /access add <upn>, /access remove <upn>, and /access list. In group chats every participant must be approved, and the agent only chimes in when actually addressed — so it stays quiet during side conversations.
Tracks open items — Captures every commitment that requires follow-up — any time someone agrees to look into something and report back. These are often small, easily-forgotten items like "Amanda will file a bug for that," or "can you revise the wording on this Figma screen?" The agent reacts to the message it captured with a 📌 emoji, even when the commitment surfaces in a side conversation it wasn't directly part of. Owner, description, status, and ETA persist between sessions, so you can ask later who's on what and the agent remembers.
On-demand workstream summary — Run /workstreamsummary run and the agent posts a digest of every open work item grouped by owner. A natural starting point if you want to graduate it into a recurring daily or weekly digest.
Workstream Q&A — Answers questions about the workstream using prior conversation history plus any additional sources you grant it access to, like your team's SharePoint site, specs, or Azure DevOps.
ough setup: how to grant
managers can run
ss to others with /access
the agent only chimes in
into something and report
ording on this Figma
side conversation it wasn't
t and the agent remembers.
n work item grouped by
onal sources you grant it
News for you
Scott Pelley fire...
Search
12:11 PM
6/3/2026

최종 정리


Microsoft Agent Platform
Build in GitHub. Run and optimize in Foundry. Reach users in M365, Teams, and everywhere work gets done.
Governed agent lifecycle
Build in GitHub
Code
Prompts
Skills + tools
Eval definitions
Run in Foundry
Optimize models, prompts, skills, and tools.
Agent runtime
plan / act / observe
host & scale
Evaluate & Optimize
routing + RL
agents & eval signals
AI Models – 11K models
Context with IQ – your enterprise knowledge
Powered by Azure – global scale, data zones, sovereign controls
Manage with Agent 365 – Observe, Secure, Govern, with Agent ID
Distribute to users
M365 Copilot
Teams
Apps + APIs

Microsoft Foundry로 오픈클로 멀티에이전트 시스템 구축해 보기

세션 링크

데모 링크