

Chat with AI in PowerShell using AIShell

Chat with AI in PowerShell using AIShell

Sean Wheeler

PowerShell Content Wrangler
Microsoft

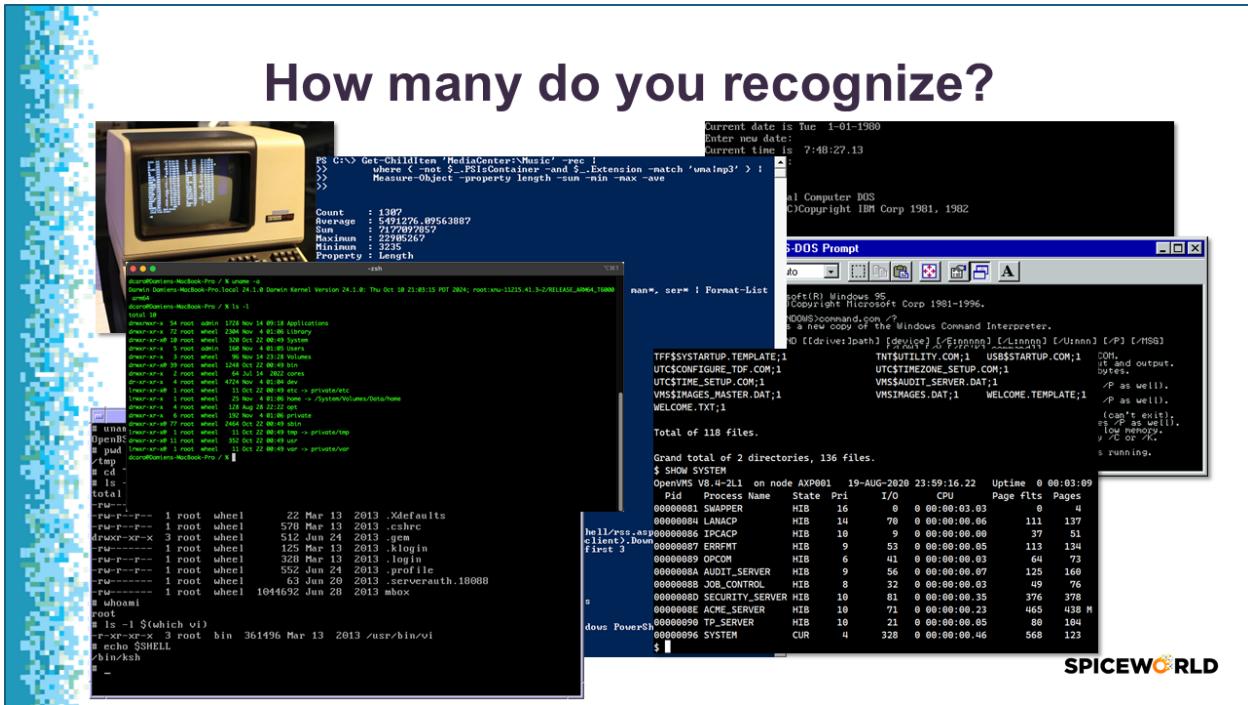
Level: Beginner

SPICEWORLD

Come learn about AIShell, the newest AI user experience brought to you by the PowerShell team. AIShell integrates with your PowerShell session and provides easy access to many different AI agents. Chat with AI from the PowerShell command line or in AIShell and post the resulting solutions back into your command line. Use AIShell to write PowerShell and Azure CLI code and explain how it works. Connect AIShell to your own language models trained with your own data. Integrate with MCPs to let AI perform actions for you.

This is a beginner session that shows you how to get started with AIShell.

How many do you recognize?



Lots of command line interfaces

The command line interface was the standard for over 30 years, until the graphical user interface (GUI) debuted with Windows and Macintosh. Today, command line interfaces (shells) are much more powerful. They allow you to automate tasks that must be done manually in the GUI. But with that power come complexity.

Observations

The prompt is intimidating

Exponential growth of commands for Clouds

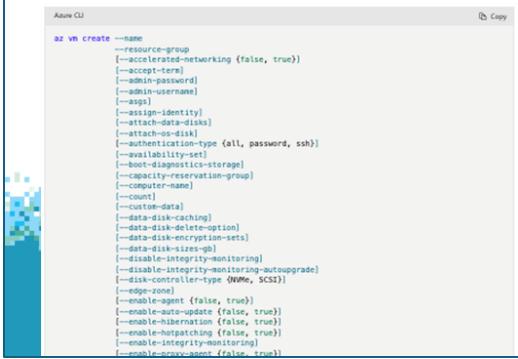
- Azure PowerShell: **6,000+ cmdlets**
- Azure CLI: **10,000+ commands**
- Microsoft Graph: **50,000+ cmdlets**

Tab completion is not always helpful

- Parameters require complex objects
- Too many commands with the same beginning

Combinations of parameters too complex

- Optional vs required parameters
- Difficult to identify value types



Azure CLI command line interface showing the 'az vm create' command with numerous optional parameters listed. The parameters include: --name, --resource-group, --accelerated-networking, --accept-term, --admin-password, --admin-username, --asgs, --assign-identity, --attach-data-disks, --availability-set, --authentication-type, --availability-set, --boot-diagnostics-icss-storage, --capacity-reservation-group, --computer-name, --count, --current-data, --data-disk-caching, --data-disk-delete-option, --data-disk-encryption-sets, --data-disk-label, --disable-integrity-monitoring, --disk-controller-type (NVMe, SCSI), --enable-ahci, --enable-e-agent, --enable-auto-update, --enable-hibernation, --enable-local-accounts, --enable-integrity-monitoring, and --enable-snmp-sensor.

SPICEWORLD

Command-line tools are hard to use

Today, the prompt can be intimidating. There are so many more commands available, with so many parameters. It's hard to know which commands to use, with which parameters, in what sequence.

Linux commands are notoriously terse. And there is no consistency with how parameters are handled across commands:

- Do I use /, -, --?
- What input types are expected?

AI Shell capabilities

Cross platform

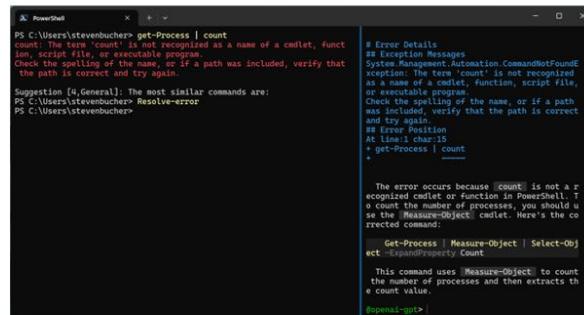
- Windows and Mac OS
- Linux not supported

User experience

- Sidecar meta language
- Interactions between sidecar and primary shell
- Suggests commands to primary shell
- PowerShell output and errors available to the agent in the sidecar

Client for various AI Agents and MCPs

- OpenAI-GPT
- Copilot for Azure
- Extensible – create your own



A screenshot of a PowerShell window titled "PowerShell". The command entered is "PS C:\Users\stevenbucher> get-Process | count". The output shows an error message: "get-Process is not a cmdlet, function, script file, or executable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again." Below the error, a suggestion is provided: "Suggestion [4,General]: The most similar commands are: PS C:\Users\stevenbucher> Resolve-error PS C:\Users\stevenbucher>". The right pane of the window displays detailed error information, including exception messages and error positions, along with a corrected command example: "Get-Process | Measure-Object | Select-Obj ect -ExpandProperty Count". At the bottom, there is a note about the error occurring because "count" is not a recognized cmdlet or function in PowerShell.

AI Shell features

AI Shell is an interactive shell that provides a chat interface with language models. The shell provides agents that connect to different AI models and other assistance providers. Users can interact with the agents in a conversational manner.

The AI Shell project includes:

- A command-line shell interface (aish)
- A framework for creating AI agents and other assistance providers
- Integration with Windows Terminal and iTerm2 on macOS
- A PowerShell module for integration with PowerShell. For more information, see the [AI Shell module](#).
- Support for MCP servers and tools
- Support for [Foundry Local](#) deployments

Each AI assistant is known as an agent. The initial release of AI Shell includes two agents:

- **Azure OpenAI** agent that connects to an instance of **gpt-4o**. Use this agent for general AI tasks.
- **Copilot in Azure** agent that can assist with Microsoft Azure knowledge. Use the Azure agent for assistance with Azure CLI and Azure PowerShell commands.

See [What is AI Shell?](#)

Install AIShell

Requirements

- PowerShell 7.4.6+
- PSReadLine 2.4.3-beta3+

Windows

- Windows Terminal

Macintosh

- macOS v14 Sonoma or higher
- iTerm2 and Python 3.11+ to use the split-screen

Installation

- Use the install script for best results
- Install the AIShell module from the PSGallery as needed (macOS)
- Make sure the version of the module matches the version of AIShell

<https://aka.ms/AIShell-Docs>

Install AI Shell

System requirements

All platforms

- PowerShell 7.4.6 or higher
- PSReadLine v2.4.2-beta2 or higher

Windows 11 or higher

- Windows Terminal

macOS v14 Sonoma or higher

- Default macOS Terminal app for the standalone experience
- iTerm2 required to use the sidecar experience
 - Python 3.11 or higher required to support iTerm2
 - Must enable **Python APIs** in the iTerm2 settings

Install AI Shell

```
Invoke-Expression "& { $(Invoke-RestMethod 'https://aka.ms/install-aishell.ps1') }"
```

On Windows, this script:

- Installs **aish.exe** to `$env:LOCALAPPDATA\Programs\AISShell` and adds it to your **PATH**
- Installs the **AISShell** module to your module path location

On macOS, this script:

- Installs the **aish** executable to `/usr/local/AISShell` and creates a symbolic link to `/usr/local/bin/aish`.

See [Install AI Shell](#)



OpenAI Compatible Provider

Configuration

- Define multiple GPTs
- Each GPT can have different model, system prompt, endpoint, etc.
- Select the one you want to use

- Generic AI assistance within the shell
- Connect the AI provider of your choice
 - OpenAI
 - Azure OpenAI
 - Gemini
 - Grok
 - Ollama
 - And more!

SPICEWORLD

OpenAI Agent

The OpenAI agent can connect to a public OpenAI service, a private deployment of the Azure OpenAI service, or any other OpenAI-compatible service. We recommend using a deployment of Azure OpenAI deployment for enhanced security and privacy. Some of these models are open-source tools for running SLMs and LLMs locally. The OpenAI agent supports the following 3rd-party models:

- [Ollama](#)
- [LM Studio](#)
- [Deepseek](#)
- [LocalAI](#)
- [Google Gemini](#)
- [Grok](#)
- [Foundry Local](#)

See [OpenAI agent](#)



Azure AI Foundry

(also known as Azure OpenAI)

- Manageability
 - Selection of models
 - Content safety and security
 - Quota management
- Train the model with **your** data
- Ensure your data stays with you
- Key based OR EntralID based Auth

Set input filter What are these categories?

Content will be annotated by category and blocked according to the threshold you set. For the violence, hate, sexual, and self-harm categories, adjust the slider to block content of high, medium, and/or low severity.

Category	Media	Action	Threshold
Violence	Text Image	Annotate and block	Medium Allow Low / Block Medium and High
Hate	Text Image	Annotate and block	Medium Allow Low / Block Medium and High
Sexual	Text Image	Annotate and block	Medium Allow Low / Block Medium and High
Self-harm	Text Image	Annotate and block	Medium Allow Low / Block Medium and High
Prompt shields for jailbreak attacks	Text	Annotate and block	Jailbreak attacks will be blocked
Prompt shields for indirect attacks	Text	Off	Content will not be annotated at all

Blocklist (Preview) Select built-in or customized blocklist

Content will not be annotated at all

SPICEWORLD

Azure AI Foundry

[Azure AI Foundry](#) is a unified Azure platform-as-a-service offering for enterprise AI operations, model builders, and application development. This foundation combines production-grade infrastructure with friendly interfaces, enabling developers to focus on building applications rather than managing infrastructure.

Azure AI Foundry unifies agents, models, and tools under a single management grouping with built-in enterprise-readiness capabilities including tracing, monitoring, evaluations, and customizable enterprise setup configurations.

See [What is Azure AI Foundry?](#)



Copilot in Azure

Beyond search

- Command and usage guidance
- Set your parameter values
- Completers for retrievable values: locations, resource group names, SKUs
- Knowledge of end-to-end scenarios using best practices

Responsible AI

- No personal or company data is stored or reused with the @Azure agent
- We only save data when you choose to share with Microsoft

```
@azure> How do I create a key vault ?  
To create a Key Vault in Azure using the Azure CLI, you can use the following command:  
  
az keyvault create --name <keyVaultName> --resource-group <resourceGroupName> --location <location>  
  
Replace the placeholders with your specific values:  
+ <keyVaultName> : The name of your Vault.  
+ <resourceGroupName> : The name you want to assign to your Key Vault. Recommended pattern: 'prod>kv<env>[-id>]'  
+ <location> : The Azure region where the Key Vault will be located.  
Run /replace/ to get assistance.  
  
--- Input Values ---  
1. az keyvault create --name <keyVaultName>  
The name you want to assign to your Key Vault.  
Recommended pattern: 'prod>kv<env>[-id>]'  
<keyVaultName>: prod-kv-1234  
2. az keyvault create --resource-group <resourceGroupName>  
The name of the resource group where the Key Vault will be created.  
<resourceGroupName>: aishell-rd  
3. az keyvault create --location <location>  
The Azure region where the Key Vault will be located.  
<location>: southcentralus  
<2/50>  
> eastus  
> southcentralus  
> westus2  
> westus3  
> australieast  
> southeastasia  
> northeurope  
> ...</2/50>  
  
--- Summary ---  
The following placeholders will be replaced:  
<keyVaultName> : prod-kv-1234  
<resourceGroupName> : aishell-rd  
<location> : westus3  
  
--- Regenerate ---  
Query: How do I create a key vault ?  
To create a Key Vault in Azure using the Azure CLI, you can use the following command:  
  
az keyvault create --name prod-kv-1234 --resource-group aishell-rd --location westus3
```

Copilot in Azure

Microsoft Copilot in Azure is an AI-powered tool to help you do more with Azure. With Microsoft Copilot in Azure, you can gain new insights, discover more benefits of the cloud, and orchestrate across both cloud and edge. Copilot leverages Large Language Models (LLMs), the Azure control plane, and insights about your Azure environment to help you work more efficiently.

Microsoft Copilot in Azure can help you navigate the hundreds of services and thousands of resource types that Azure offers. It unifies knowledge and data across hundreds of services to increase productivity, reduce costs, and provide deep insights. Copilot in Azure helps you learn about Azure by answering questions, and it can provide information tailored to your own Azure resources and environment. By letting you [express your goals in natural language](#), Copilot in Azure simplifies your Azure management experience.

See [Microsoft Copilot in Azure Overview](#)

The Power of **Copilot in Azure** wherever you work



Azure Portal



AI Shell



GitHub Copilot
for Azure

SPICEWORLD

Copilot in Azure everywhere you are

Microsoft Copilot in Azure is available from everywhere you want to work with Azure from.

- Azure Portal
- AI Shell
- GitHub Copilot for Azure in Visual Studio Code

You get the same Copilot for Azure capabilities in in location.

See [Microsoft Copilot in Azure capabilities](#)



Model Context Protocol (MCP) Support

- Open protocol that enhances the capabilities of an LLM
- Provides commands that give access to external data to enhance context
- Provides commands that can perform actions on your behalf

SPICEWORLD

MCPs for automation

Model Context Protocol (MCP) is an open protocol that standardizes how applications provide context to large language models (LLMs). Starting in AI Shell 1.0.0-preview.6, AI Shell can act as an MCP Host and client to MCP servers. The key participants in the MCP architecture are:

- MCP Host - AI Shell coordinates and manages one or multiple MCP clients
- MCP Client - The client component of AI Shell maintains a connection to an MCP server and obtains context from an MCP server for the MCP host to use
- MCP Server - A program, running locally or hosted remotely, that provides context to MCP clients

MCP tools enable AI agents to access external tools and services to enhance their capabilities and provide more accurate responses. MCPs can integrate with various APIs, databases, and other resources, allowing agents to retrieve real-time information and perform complex tasks.

See [MCP support](#)

Public preview

- Try it: <https://aka.ms/AIShell>
- Open Source: <https://github.com/PowerShell/AIShell>
- Known limitations
 - Not supported in elevated mode on Windows Terminal
 - No side car support for Linux
 - It's a preview 😊 may be buggy



SPICEWORLD

Try it out

Try it out: <https://aka.ms/AIShell>

Open Source: <https://github.com/PowerShell/AIShell>

Known limitations

- Not supported in elevated mode on Windows Terminal
- No side car support for Linux (and not tested)
- It's a preview 😊 so it may be buggy

Thanks for coming!

Connect with me

- <https://seanonit.org/docs>

Fill out your session surveys



SPICEWORLD

Presented by Sean Wheeler

Check out my website for links to my social media and other topics I have presented.

Presentations: <https://seanonit.org/docs>

GitHub: <https://github.com/sdwheeler>

Bluesky: [@sdwheeler.bsky.social](https://bsky.social/@sdwheeler)

LinkedIn: [in/scriptingsean](https://www.linkedin.com/in/scriptingsean)